Efficient Utilization of Big Data using Distributed Storage, Parallel Processing, and Blockchain Technology

Alessandro Giuliano [a], Waleed Hilal [a], Naseem Alsadi [a], Onur Surucu[b], S. Andrew Gadsden [a], John Yawney [a,b], and Youssef Ziada [c]

[a] McMaster University, 1280 Main St. West, Hamilton, ON, Canada, L8S 4L8;
[b]Adastra Corporation, 200 Bay St., Toronto, ON, Canada, M5J 2J2;
[c]Ford Motor Company, 1 American Rd., Dearborn, MI, USA, 48126

ABSTRACT

As data collected through IoT systems worldwide increases and the deployment of IoT architectures is expanded across multiple domains, novel frameworks that focus on application-based criteria and constraints are needed. In recent years, big data processing has been addressed using cloud-based technology, although such implementations are not suitable for latency-sensitive applications. Edge and Fog computing paradigms have been proposed as a viable solution to this problem, expanding the computation and storage to data centers located at the network's edge and providing multiple advantages over sole cloud-based solutions. However, security and data integrity concerns arise in developing IoT architectures in such a framework, and blockchain-based access control and resource allocation are viable solutions in decentralized architectures. This paper proposes an architecture composed of a multilayered data system capable of redundant distributed storage and processing using encrypted data transmission and logging on distributed internal peer-to-peer networks.

Keywords: Big Data, Blockchain, Edge, Fog, Cloud, IoT, IPFS, Apache Spark

1. INTRODUCTION

In recent years, the Internet of Things (IoT) has been a heavily researched field in both industry and academia. More and more objects are being equipped with hardware capabilities to connect and exchange data through various communication technologies like Bluetooth, Wi-Fi, and mobile networks such as 4G and 5G. Research from Trasforma Insights has shown that the number of IoT-connected devices operational worldwide could reach 25 billion by 2030. This push towards digitalizing physical objects has exponentially increased the collected, stored, and processed data. The combination of the increase in computing power of the hardware-enhanced data collection and storage systems and better data science algorithms and techniques provide great opportunities in the development of automated systems for optimal policy decisions, the implementation of predictive corrective actions and in general has provided more significant insights in the monitoring of systems across domains.

Cloud technology has revolutionized how companies store and process their data over the last ten years with the creation of new business models such as software as a service (SaaS), infrastructure as a service (IaaS), and platform as a service (PaaS). These cloud solutions leverage cutting-edge technology for the management of data, employing redundant distributed file systems and parallel processing; examples of commercially available services include Google file system (GFS)[1], Amazon elastic file system (EFS), Microsoft Azure distributed file system (DFS). Many open-source projects have also been developed to enhance further distributed file system technology, such as the Hadoop distributed file system (HDFS) [2] and the Inter-Planetary file system (IPFS) [3]. However, cloud computing has its limitations. Given the physical distance of the cloud servers to the source of the data, the transmission of data over the internet can often have high latency and low response time, rendering the use of cloud technology unfeasible for specific applications [4]. To cope with this issue, the development of new paradigms such as edge computing, fog computing, cloudlets, and multi-access edge computing (MEC) paradigms has opened new doors for creating novel IoT architectures that can support real-time processing. Edge and Cloud computing effectively complement each other in latency and response speed versus computational and storage capacity. Furthermore, work to enhance the interconnectivity of edge devices (resources

Big Data IV: Learning, Analytics, and Applications, edited by Fauzia Ahmad, Panos P. Markopoulos, Bing Ouyang, Proc. of SPIE Vol. 12097, 1209704
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2618891

deployed in near proximity of where the data is collected) has been brought forward by research through the creation of new protocols for machine (M2M) communication in IoT systems such as the libp2p open-source project, on which Ethereum Blockchain 2.0 builds off.

The popularization of blockchain technology through new cryptocurrencies such as Bitcoin, Ethereum, and Cardano has also opened the door to creating new decentralized architectures. Moreover, smart contracts have been popularized by modern non-fungible tokens (NFTs) and introduced a novel way to recognize proprietorship or keep track of virtual objects using blockchain. These technologies' reach goes far beyond current applications; they can be used across domains ranging from financial markets to logistic and industrial applications. An example of current open-source projects to develop a multipurpose blockchain is Enterprise Ethereum, R3's Corda, Hyperledger Fabric, Quorum, OpenChain, Multichain, and more as new projects are born. Using a distributed storage/computing system complemented with the security of blockchain access control and a distributed cognitive engine for resource allocation offers the opportunity to create a more efficient, flexible, and secure IoT architecture. The purpose of this paper is to provide a concrete architecture design, implementable using multiple open-source project tools to create a hybrid, edge/cloud IoT framework.

The paper is structured as follows; Section II will cover some background and motivations behind this paper, Section III will cover related work in the areas of distributed IoT architectures as well as blockchain, and IoT integrated implementations, in Section IV, the architecture proposed will be outlined and discussed in all its practical components and aspects, to then conclude the paper in Section V.

2. BACKGROUND AND MOTIVATION

As technology moves forward, more and more devices will be connected to the internet via the integration of microprocessors within objects and machines of various sorts, effectively rendering them Cyber-Physical Systems (CPS) capable of storing, processing, and communicating data over the internet. This push towards a more holistic internet of things is moved by the benefits of technology integration, both for people and companies alike. However, this new wave of interconnected devices, along with the advancement of smartphone technology, also raises issues in how data is currently stored and processed. More bandwidth-hungry devices mean higher latency from Cloud servers, which are the de facto model for big data processing, resulting in a reduced quality of service (QoS) to the end-user. This creates an issue of scalability for IoT systems and is a limitation for IoT deployments in the application that require low latency to operate effectively, such as machine equipment monitoring, intelligent city resources management, or IoT systems related to monitoring remote patient conditions. Furthermore, cloud servers are often physically located far from where the data is collected, effectively increasing data transmission and response times [5]. Edge and fog computing were proposed to solve this problem.

2.1 EDGE AND FOG COMPUTING LAYERS

Edge computing is an architecture model that employs resources at the edge of the network, while fog computing refers to a model that employs both edge and cloud computing, managing the resources dynamically to store and process data [4]. A fog model architecture generally has three layers. The first layer is the edge layer, where CPS interacts with the environment, collecting information through sensors and actively modifying it through actuators. The second layer is the fog layer, which is near the first layer and is composed of servers dedicated to storing and processing some of the data with constraint capabilities in terms of resources. Finally, the third layer is the cloud, where highly virtualized and parallelized servers carry out more intensive data processing. A schematic of a layered fog architecture can be seen in Figure 1 below.

Cloudlets are small-scale cloud datacenters used for fog computing. They are based on the same principles of cloud technology, including virtualization, distributed mass storage, and parallel programming while also having limitations in hardware resources. Virtualization technology can be deployed in various forms, such as virtual machines (VMs) and containers. However, fundamentally, it is creating operating system (OS) instances that share the same hardware

and kernel. In contrast, resource allocation and creating such guest VMs are managed by a computer software called a hypervisor or virtualizer compared to an emulator. A visual diagram of such a schematic can be seen in Figure 2.

Furthermore, virtualization allows creating virtual clusters (VCs), which emulate multiple nodes, capable of parallelizing tasks such as storing and processing data within the same hardware, increasing resource efficiency [6]. The advantages brought by such architecture are platform independence by the replicability of such containers and VMs, resource abstraction since these OS are not run on bare metal hardware, and isolation such that if one of the instances fails, there are plenty more that can carry on with their tasks, hence creating redundancy withing the operating server [7]. However, despite the advantages, virtualized edge/fog servers and cloudlets also have some drawbacks in terms of limited resources, limited-service range, lower reliability of single instances, and vulnerability to cyber-attacks.

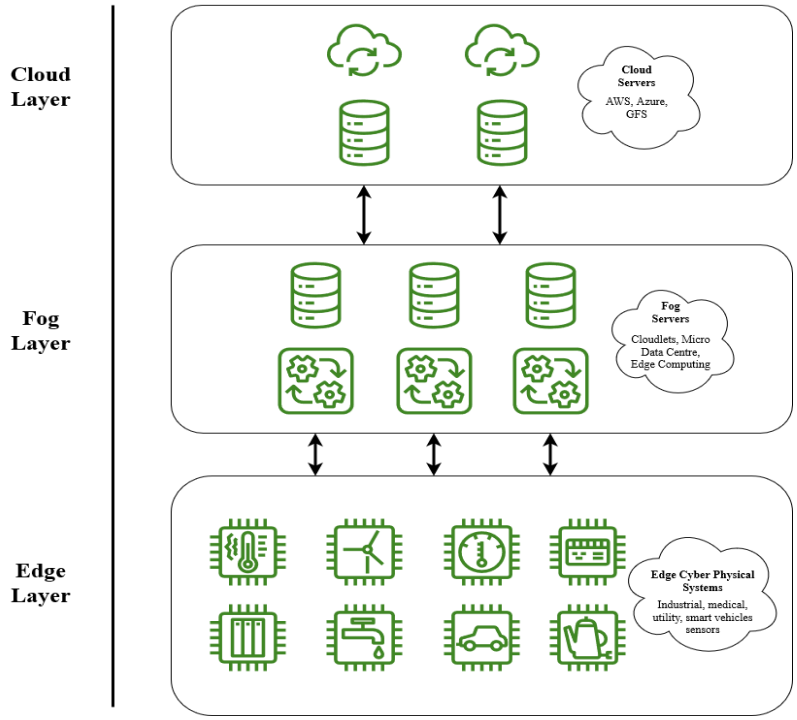

Figure 1: Three-Layered Fog Computing Structure Diagram

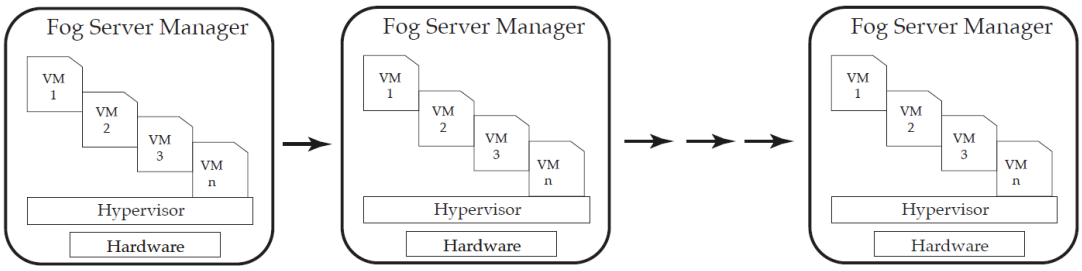

Figure 2: Edge/Fog Virtualized Schematic, adopted from [5]

As mentioned, the limited hardware capability severely limits the amount of storage and processing capabilities of such servers; for this reason, many companies and research groups are working towards innovating virtualization software through the creation of more lightweight virtualization techniques in order to minimize the computational expense of having multiple OS running at the same time. Examples of such efforts in recent years are the introduction of Multipass by Canonical Ltd., the parent company of Ubuntu OS, which offers a more efficient virtualization engine for the creation of Ubuntu instances and improvements in the existing virtualization software by researchers such as Xen, Linux KVM and OKL4 Micro visor [8]. Furthermore, load balancing is also necessary to optimally process high amounts of data traffic. Load balancing is the process of offloading tasks across a sub-set of fog nodes or clusters, decreasing the number of tasks per node by spreading the load across the fog layers. Offloading to cloud servers is also required for data processing and storage that cannot be met in the fog layer. The orchestration of such a system could be done using a centralized server control or through a decentralized cognitive engine.

2.2 CENTRALIZED VS DECENTRALIZED DISTRIBUTIONS

Centralized and decentralized architectures offer advantages and drawbacks on crucial aspects of IoT architecture design. The offloading criteria for an efficient orchestration can be based either on application characteristics such as computation and communication demands, latency sensitivity, or on edge-cloud resource availability such as resource utilization (nodes CPU, ROM, and RAM utilization) and resource heterogeneity (difference in hardware capabilities of each node) [9]. Centralized storage and processing are based on the orchestration carried out by a node referred to as the master node and can be based on various key aspects of the system while not impacting the operation of nodes employed on the edge/fog. Centralized systems are widely covered in literature due to their simplicity of implementation but face extreme scalability and security issues, subject to single point failure (SPOF) [4]. It must be noted that there are solutions to the single point failure through transparent application failover, which essentially relies on a dormant copy of the master to be activated if the master goes down. However, it is still inferior to a decentralized and distributed file system redundancy.

On the other hand, while still focusing on the same criteria, decentralized orchestration is carried out by every node or node cluster in the lightest configuration possible, offering significant redundancy compared to centralized solutions subject to single-point failures. Research in this area is still limited, but some proposed implementations are based on game theory and the concept of Nash equilibrium [4]. This consideration between centralized and decentralized solutions can also be extended internally in each fog cluster by choosing different storage and processing solutions commercially available. For instance, Hadoop is a centralized storage system in managing the data, relying on a master node to track where the data is stored (using a hash table) [2]. At the same time, IPFS is decentralized, not having a centralized record of where the data is stored and instead relying on P2P communication to retrieve such data using Merkle DAGs [3]. Section IV, architecture design, will discuss how these storage solutions operate.

Being a virtualized environment much like cloud, fog servers are subject to security threats such as access control issues (ACI), data loss (DL), data breaches (DB), insecure API (IA), malicious insider (MI), advanced persistent threats (APT), eavesdropping, jamming and more [5] through various attacks such false data injection (FDI) and man in the middle attacks (MIM). Therefore, security concern arises in the development of edge/fog architectures. In general, many modern IoT architectures lack security considerations to maintain the CIA triad, confidentiality, integrity, and availability. Especially in the industrial IoT and manufacturing sector, several security issues emerge as these systems were developed with the assumption of security by isolation, running on a local network, and being shielded from the internet through firewalls. A notable example is the Stuxnet virus effect on the SCADA monitoring systems of Iranian manufacturing facility PLCs in 2010 [10]. Blockchain integrated with IoT has been proposed to enhance IoT systems' security, leveraging encryption techniques that are part of the blockchain's natural design.

2.3 BLOCKCHAIN AND SMART CONTRACTS APPLIED TO IOT ARCHITECTURES

A blockchain is a distributed network of peers or nodes in a trustless peer-to-peer environment. Every node or actor shares the same copy of a shared ledger that contains a record of all transactions registered in the blockchain. It has been popularized through cryptocurrencies such as Bitcoin and Ethereum, but its application extends across domains. Fundamentally blockchain can maintain a reliable registry of transactions in a trustless network using two core features, encrypted keys and a consensus mechanism. Every actor on the blockchain retains a public key and a private key. Public keys are used to recognize a node publicly, while private keys are used to sign transactions, such as sending cryptocurrencies to another wallet; a visual representation of this mechanism can be seen in Figure 3. The use of hashing algorithms such as SHA-256 and Blacke-256 ensures that once a transaction is signed, the reverse engineering of the private key is virtually impossible [11]. Practically once a transaction is signed, the message is hashed using the private and public key, giving out a unique identifier that mining nodes can verify the legitimacy of the transaction. Mining nodes are actors that collect the broadcasted recent transaction collecting them into blocks and verifying every transaction to be acceptable; this requires high computation given the hashing algorithms used and are incentivized by the blockchain through rewards in terms of cryptocurrency. The consensus algorithm is used to task the creation and addition of the mined blocks to the blockchain, making the overall blockchain agree on the next block to be added to the chain. The consensus protocol is crucial in a distributed network; it affects the performance of the blockchain and is the guarantee for the stable operation of blockchain systems. Many different consensus mechanisms have been developed for different blockchain applications such as Proof of Work, Proof of Stake, Delegated Proof of Stake, and Proof of Authority.

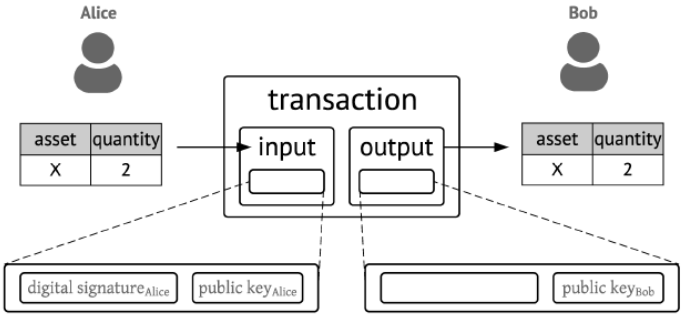

Figure 3: Blockchain Transaction Visual Representation [11]

Furthermore, smart contracts have been introduced by Ethereum 2.0 and other blockchain developer companies to expand the utilization of blockchain technology not only to transact digital currencies but also digital assets registered on the blockchain. This feature is especially relevant to applying blockchain to IoT; by creating a digital record of physical goods, it is possible to effectively maintain a register of the proprietorship of goods and their previous track record. For instance, blockchain and smart contracts have been proposed as a novel way to keep a record of transactions in the logistics of a given transportation company; a typical example is the use of such system applied to grocery items, in such system the consumer can access the record of the good through a user-friendly app, and access information such as when and where the product was cultivated, picked and brought to the grocery store. This example applies to many IoT architectures, such as smart grids [12], but it also branches to virtually anything that needs a log of transactions. This paper will apply this concept to regulate access management to specific files stored in distributed file systems.

3. RELATED WORK

Multiple papers have been published in recent years that discuss the development of new edge and fog computing architectures, as well as discussion regarding the integration of blockchain to IoT systems. Most touch upon specific parts of the overall structure while others tackle it from a more general point of view, envisioning the future smart cities and smart factories.

Yang et al. present a survey of the research issue and challenges of integrated blockchain and edge computing systems [13]. Making a point of how edge computing development is challenged by severe security challenges and technical challenges in the scalability of such systems. Among these, it explains the limitation of blockchain scalability faced by current solutions such as low throughput, high latency, resource exhaustion comparing the performance of Ethereum and Bitcoin to the current state of the art payment systems such as Visa credit card processing, which can achieve over 2000 transactions per second on average [13], and bridging the discussion to IoT architectures. It also covers the benefit of the integration of blockchain in edge computing, namely the enhancement of security, privacy, and automatic resource usage, and discusses such integration requirements. The essential requirements outlined are authentication, the requirements of entities recorded on the blockchain, the adaptability of such integrated systems, network security, data integrity, low latency, and verifiable computation (when outsourced). It presents a general framework of a private blockchain based on a local network consisting of edge and fog nodes, touching upon P2P communication for M2M communication IoT systems and secure multiparty communication. Moreover, discussing such a system's storage, processing, and network components[13].

Nyamtica et al. present a practical approach in implementing decentralized blockchain storage systems giving a conceptual analysis of the requirements for scalable and secure blockchain data storage in edge computing platforms within a broader IoT architecture [14]. Namely, the authors recognize offloaded computation, decentralized data storage, authenticated transactions, anonymity, controlled access, the integrity of data, low latency, and adaptability as the target requirements for implementing a hybrid blockchain/edge computing framework. Recognizing anonymity, integrity, and adaptability as critical issues. It also emphasizes the need to outsource computation to external actors closer to the end-user, specifically the nearest, more capable peer, optimal response time[14]. The presented architecture not only relies on a cluster of private peers but also attempts to leverage public network peers to boost the computational power and reduce latency.

Confais et al. propose a distributed IoT implementation in Fog/Edge computing using IPFS and scale-out network-attached storage system (NAS), a Bit-Torrent based object store presenting test results for reading writing speed of the system[15]. This architecture is based on micro centers placed close to the edge of the network enabling low latency response in computation and storage of user files. The paper surveys multiple distributed storage solutions like Rados, Cassandra, and IPFS, which are chosen to be tested on the Grid5000 experimental setup. Different implementations of IPFS paired with RozoFS as the Scale-out NAS are tested. Results show that the performance of local functions is not impacted by the overhead of the Scale-out NAS system, instead of showing an improvement of 34% in access time [15].

Mehbodniya et al. present a bilevel fog/cloud architecture that leverages blockchain for security and privacy in a smart healthcare setting [16]. The system is comprised of two main segments, the first responsible for authentication and authorization of new patient registration and new medical devices registrations. These are health monitoring sensors that continuously monitor the patient's vital parameters depending on their case situation to relay this information to the system. The second segment is the dissemination of such data and metadata through the blockchain network. Data is preprocessed at the edge nodes to be transmitted to the cloud for further analysis, using an IPFS cluster storage solution. Smart contracts and private blockchain are used for access control, leveraging a smart contract created to maintain a record of the authorized nodes, and the blockchain ledger is used to keep a record of data access and transmission [16].

Deepa et al. present a novel architecture based on the Edge of Things paradigm paired with blockchain technology, which primary purpose is to enable future low-latency and high-security services and applications [17]. The framework comprises three layers: industrial application, MEC, and IoT. The blockchain is deployed across all three layers and ensures access authentication, data privacy, attack detection, and trust management. The paper also covers possible applications of such BEoT architecture applied to healthcare, smart grids, transportation, and smart homes[17].

Rahman et al. present a vision for the smart city of the future, relying on a cognitive edge of things paradigm complemented with the use of public blockchain networks, in contrast with most of the literature that leverages private implementation of blockchain technology[18]. It focuses on the scale of future smart cities and the challenges in collecting and utilizing large amounts of data to predict, alert, and prevent adverse events within the city. The paper envisions a shared economy based on blockchain for booking, transacting, and sharing resources throughout the city, highlighting how smart contracts could replace a central verification authority. The system proposed to leverage RESTful architecture to read and write IoT data based on MEC servers [18].

4. ARCHITECTURE DESIGN

So far, the need and motivation for a hybrid distributed fog cloud architecture was outlined, and previous work on the subject, challenges in the deployment of such a system were also presented in terms of security and efficiency. In this Section, a novel hybrid architecture will be presented, integrating distributed storage and processing systems, resource allocation management, and blockchain to utilize big data efficiently. First, a bottom-up approach will be used to describe each layer of the system; then, the multilayer components will be described, in this case, the blockchain distributed ledger and cognitive control system. The main constraints to implement the proposed architecture must be defined a priori to the breakdown of each architecture component and are as follows.

- Edge and fog servers must be highly virtualized to enable distributed storage and parallel processing.
- A VPN to create an intertwined local network is unfeasible in the proposed framework as it would dramatically increase the latency between edge, fog, and cloud servers.
- The system cannot rely on centralized resource allocation as it would defy the principle of decentralization upon which the architecture builds.
- Fog servers must be in relative proximity to the edge servers, at least within the same state or province; the closer, the better.
- Data transmission among layers must be encrypted for security purposes using end-to-end encryption.

The first layer, the edge layer, is the only cyber-physical layer composed of sensors and actuators that interact with the control environment. Depending on the application, these will vary; in an intelligent factory scenario, a practical example can be cameras, temperature sensors, humidity sensors, and additional sensors embedded within manufacturing equipment that return process parameters such as voltage, current, wire feed rate, gas flow rate and more in the case of a robotic welding process. The same applies to other domains with differences in the sensor's types. Although usually, a common factor is the heterogeneity of the data types, which is an issue when trying to extrapolate semantic knowledge for corrective action or optimal policy decisions. In the Industrial IoT (IIoT) case data collected in this layer is stored in nearby servers, often these are servers bridged onto the same local network, internal to the plant. These can be considered edge servers as they reside at the very edge of the network with the sensors. Data is generally offloaded to SQL servers that are run within the plant to then be processed and then offloaded to cloud services if the amount of data is too high or if the computation hardware requirements are unable to be met by local servers. This changes significantly in a smart city scenario as the data is offloaded to nearby data centers that cannot be run on a local setting unless using a VPN. Using a VPN would dramatically increase the latency as data has to be relayed to a remote server and then rerouted to the data center; hence it is unfeasible in the framework.

The second layer, the fog layer, comprises data centers in the proximity of the edge; these are highly virtualized environments composed of clusters. These cloudlets virtualize hardware resources, making them available to the cluster for VMs, leveraging a hypervisor to distribute computing power. Each data center will run on a local network and comprise multiple clusters of virtual instances. To distribute such cloudlets, docker is used to employing replicas of virtual instances across the edge/fog data centers. This makes the deployment and update of VMs across servers more accessible and practical; the upgrades will be distributed using cloud requests through a deployment manager. In addition, these clusters can interact with other nearby clusters to share computational and storage resources.

The third layer, the cloud layer, is physically distant from the edge layer and takes on ample storage and processing of big data using computationally expensive algorithms such as machine learning algorithms. The computation performed on the cloud cannot be used for real-time feedback control and is carried on in batches. This computation will aid the cognitive controller to extrapolate higher-level knowledge and will be used to assess the state of the environment or the system. It will also be used to perform data analysis to report the current states at a higher level give network managers the ability to create periodic reports of current states. These reports can vary in type and be technical, depending on how they will be used. They span from more technical to less technical and more business-related reports. A visual representation of the overall layered structure of the architecture can be seen in Figure 4.

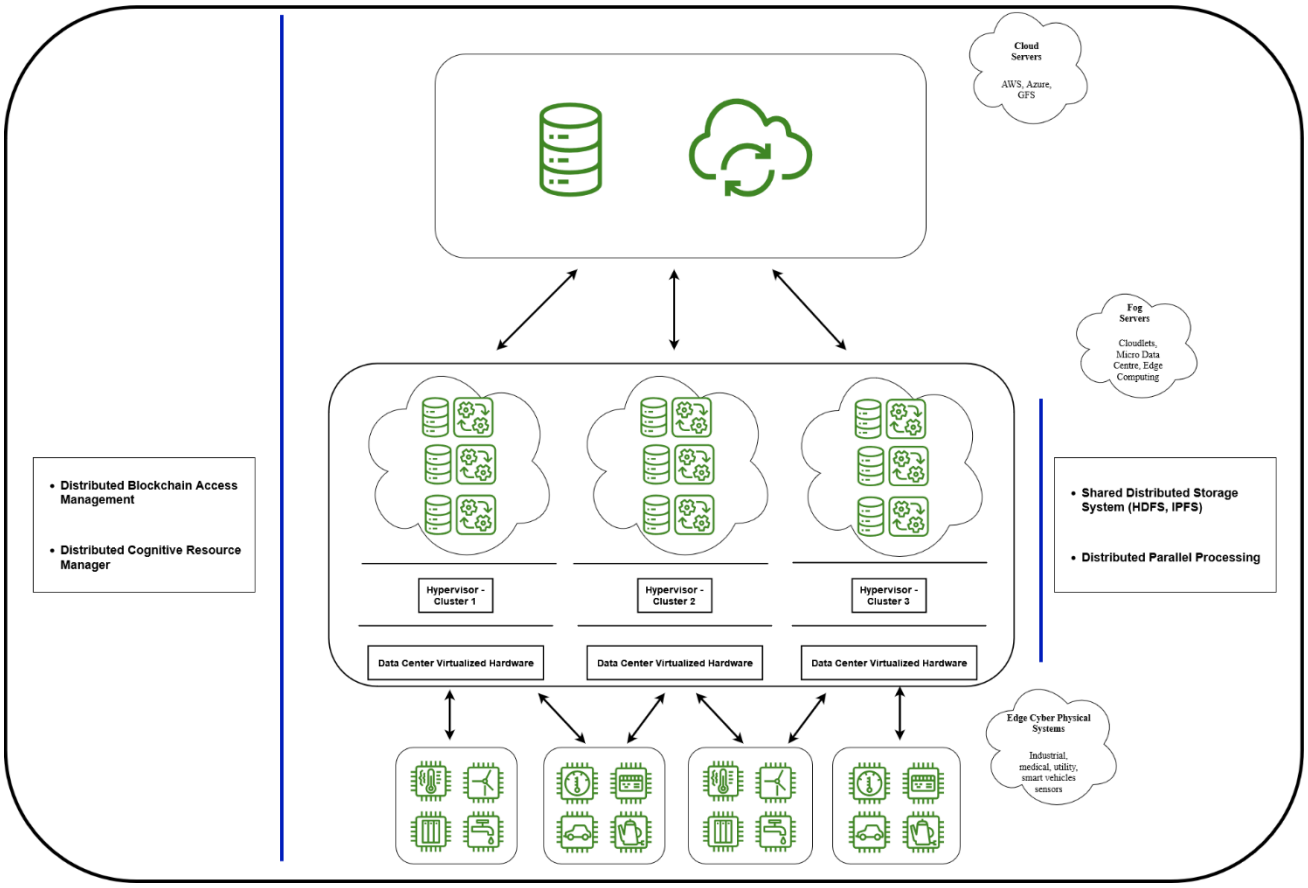

Figure 4: Visual Representation of Proposed Architecture

4.1 DISTRIBUTED STORAGE

Various distributed storage solutions are available in the market to be employed in the proposed architecture. The most notorious is Apache HDFS open-source project. Hadoop is an overall distributed storage system; it includes many useful features to ensure the integrity and reliability of data storage, such as sharding and redundancy features. Sharding is the process of breaking up files into smaller pieces called chunks of data to store them among multiple network nodes, while redundancy features ensure that these chunks are replicated and stored in multiple places within the network. In addition, these features ensure that if one of the nodes goes down or is compromised, the data is still retrievable and re-constructible to the original form. However, HDFS is not a fully decentralized system; Hadoop uses a particular node called a Master node to manage stored chunks. This node contains the complete data hash table (DHT), which is simplistically an index of file's chunks and location, such that if a node requests to retrieve a file, the master node serves as a guide to the software to where to retrieve each chunk and how to rebuild the file. Unfortunately, this makes the system prone to single point failure, although Hadoop 2.0 remedies this by creating dormient replicas of the master node that are activated if the master node goes down or is unavailable to the system. Hence making it a two-point or more failure system depending on how many dormant replicas are created throughout the network.

IPFS is another distributed file system that differs significantly from HDFS, based on the libp2p library for M2M communication. This system does not use a master node; instead, it uses the Secure Kademlia protocol [19][20], similar to gossip-like protocol, to retrieve file chunks using acyclic Merkle DAGs. Merkle DAGs are an alternative to DHT to break up files into chunks and reconstruct them; by having the root hash of any file, the system can retrieve all the shards that compose any given file from the network of nodes and verify data integrity. The main advantage of using this system is that the system can be considered genuinely redundant given its truly decentralized nature. Furthermore, the extension of

IPFS, IPFS Cluster allows the replication of shards across nodes, ensuring that if one node is compromised, the shards contained in such node can also be retrieved elsewhere in the network.

An IPFS private network is the distributed file system of choice in the architecture, given the robustness of the peer-to-peer network architecture. The various clusters share a cluster secret used as a password to bootstrap to the IPFS network of nodes. In the architecture, one private network will be deployed on the fog level, and a separate one to the cloud to avoid that shard of files meant to be stored at the fog level to be also stored on the cloud, as this would effectively compromise the concept of processing and storing the data close to the edge to have lower latency. Effectively the data will be sent to either network, fog, or cloud depending on resource usage and the type of algorithm to be run on the data. Data offloading from edge to cloud will also be used for older data and will be transferred similarly using cloud API. This also ensures a further security measure for the cloud if the fog layer is compromised.

4.2 PARALLEL PROCESSING

Parallel processing will be employed by both the fog and cloud layers to process the collected data more efficiently. Apache Spark and Apache MapReduce are among the most common software used to parallelize data computation, spreading the computational node across nodes. Apache Spark has several advantages compared to MapReduce, being more flexible and maintaining a working copy of the data processed in the RAM. More information regarding Apache Spark benefits over MapReduce can be found here [21]. Apache Spark resilient distributed dataset (RDD) allows Spark to employ several algorithms to a specific data set or a subset of data without having to fetch and distribute the data every time it must run. This makes it a lot more efficient when compared to MapReduce when performing different types of computation on the same set of data [22]. It also uses less disk space and supports higher-level operators [21].

Limited research has been carried out to integrate IPFS and Apache Spark, but through testing, the authors were able to bridge the two systems making them work together effectively; the efficiency of such system will be covered in future publications, and in this paper will be assumed that the cooperation between the storage and processing components is feasible. Apache Camel is another solution that will be further explored to make a pipeline for the data, to make IPFS and Spark communicate more efficiently. Therefore, this architecture allows parallel processing both on the cloud and the fog layer by effectively virtualizing the data centers close to the network's edge. This allows for more efficient processing on fog for real-time data analysis and allows to perform more computationally intensive algorithms given the distribution of computing power across nodes and across data centers, which are all interconnected through a private IPFS and Spark network. In such a system, it is worth noting that the closer nodes will be prioritized, and the optimal distribution of workload will have to be determined algorithmically.

Resource management for hybrid fog cloud IoT architecture is essential to ensure an efficient data processing pipe line. Many solutions have been proposed for this task, focused on different resource management challenges such as energy consumption, data management, locality, load balancing, dynamic scalability, latency sensitivity, and more [23]. In the proposed architecture, the main challenges are latency sensitivity, load balancing, and orchestration in fog/cloud IoT. Workload balance optimization algorithms aim to manage and avoid overload, congestion, and low-load resource management issues. Several can be found in the literature, such as GPRFCA, DRAM, PBRA, and ERA, a comparison of each can be found in [24]. Different optimization algorithms can be selected to make up an application specific cognitive controller depending on the domain to which the IoT architecture will be applied. A combination of resource provisioning and task scheduling algorithms should also tackle the latency and orchestration requirements in the proposed architecture. In addition to any of the mentioned algorithms, the implementation of task scheduling algorithms such as Fog Sync Differential Algorithm (FSYNC) and Reed-Solomon Fog Sync (RS-FSYNC) should be able to complement the design of a comprehensive cognitive controller for the specific application [24].

Furthermore, in distributed architectures, blockchain-based, game-theoretic approaches, genetic algorithms, and sensor function virtualization (SFV) based approaches can be undertaken to control hardware resources [4]. FocusStack is geobased orchestration architecture developed by *Amento et al.* and the AT&T Research Labs. This solution is particularly useful in resource management of edge devices that effectively move around, such as smart cars and smartphones. Furthermore, it is the best architecture for situational awareness-based orchestration of mobile devices using OpenStack to virtualize lightweight edge devices [25].

4.3 BLOCKCHAIN ACCESS CONTROL

In recent years, the integration of blockchain technology with distributed file systems and hybrid fog/cloud architecture has been a heavily researched topic. Given the security issues that arise with deploying such hybrid architecture, such as possible jamming, sniffing, man in the middle, false data injection attacks, and more, the need for a system that bridges across layers and ensures the security and trackability of data storage, computation, and network is evident [13]. The need for secure access control can be ensured using blockchain solutions, past research on the integration of blockchain into distributed file systems and IPFS can be found in the following publications [10], [11], [13], [14], [16]–[18], [26]– [33].

Some limitations must be accounted for when employing such a solution; blockchain requires increasing storage space, and the speed at which transactions are executed and the speed at which blocks are mined can decrease considerably as the size of the chain increases. This can be considered a scalability issue when developing a blockchain access control security solution. To overcome this issue, different versions of blockchain solutions can be used, either by building a big data distributed database to store transactions and blocks, where decentralized control is achieved using DNS federation or by using modified private blockchains such as Corda R3, which considerably reduces the size of the ledger by storing transaction only in a network subsystem which is comprised of the involved parties in the transaction. This ensures privacy and reduces the amount of data stored in the Hyperledger; therefore, no entity has the entire history of all the transactions in the network. Instead of mining blocks in this blockchain implementation and form, the blockchain uses neutral and trusted parties called notaries to ensure the transaction's validity. The records can be stored in an off-chain database for record keeping or requested from other nodes if needed.

An alternative solution is to use Ethereum 2.0 based smart contracts to maintain a record of all transactions. The root hash of stored files is saved in smart contracts that perform the access control in such a system. When a file must be retrieved, an intelligent contract is tasked to return the root hash, the smart contract validates the public key of the node requesting the transaction, and if it is included in the list of allowed entities, the root hash is returned to the node such that it can then retrieve the file through the IPFS storage system. This type of access control can be extended to all the components of the data flow of the architecture, read/write permission to either add a file, review or modify existing files stored in the distributed network. Moreover, by using proof of work consensus protocol and dedicated mining nodes, creating a shared ledger that keeps track of data transactions is possible and does not impact the performance of nodes dedicated to storing and processing data. A hybrid blockchain IPFS implementation was explored by *Steichen et al.* [26] using Ethereum 2.0 blockchain. The system was tested in terms of latency and stress requirements, and results show that the proposed act-IPFS achieved a slightly higher time delay in adding files to the distributed IPFS storage, although without presenting significant drawbacks in the case study presented. Stress test results aimed at determining the scalability of such a system and how many transactions can be performed, a maximum of 190 operations per second was achieved in the experiments. A graphical representation access control-based blockchain-IPFS hybrid system can be seen in Figure 5 below. Private blockchain networks can drastically increase the security of hybrid fog/cloud systems; the implementation of access control allows the distribution in a trustless peer-to-peer network of access control through smart contracts, allowing data records to be available at any node in the network.

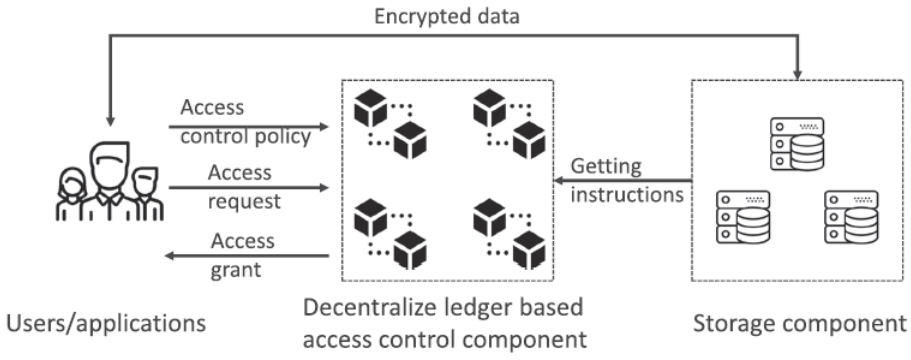

Figure 5: Decentralized blockchain access control diagram.

6. CONCLUSION

A practical hybrid fog/cloud architecture was presented using existing open-source software and blockchain technology for access control, ensuring privacy, and countering existing vectors of attacks in IoT systems. The need for these types of architectures was presented in time-sensitive applications and the means to create and deploy such systems. Vulnerabilities in the deployment of hybrid architectures were covered, and the integration of private blockchain networks was presented using available technologies such as Ethereum 2.0 and blockchain and Corda R3. Future research will focus on testing various deployments and versions of this architecture to determine efficiency and applicability to specific scenarios. Ideally, the goal is a holistic solution that is flexible enough to be applied across domains. Also, specific aspects within the architecture must be further explored in the specific resource allocation, and fog orchestration using a cognitive controller will be the focus of future research. Furthermore, given the research intensity, new blockchain solutions may be proposed in the near future, and the integration of such architectures within the proposed framework will be explored.

REFERENCES

[1] S. Ghemawat, H. Gobioff, and S.-T. Leung Google, “The Google File System,” 2003.

[2] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design," 2005.

[3] J. Benet, "IPFS-Content Addressed, Versioned, P2P File System (DRAFT 3)," 2014.

[4] C.-H. Hong and B. Varghese, "Resource Management in Fog/Edge Computing: A Survey," Sep. 2018, doi: 10.1145/3326066.

[5] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1. Springer Verlag, Dec. 01, 2017. doi: 10.1186/s13677-017-0090-3.

[6] Z. Tao *et al.*, "A Survey of Virtual Machine Management in Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1482–1499, Jul. 2019, doi: 10.1109/jproc.2019.2927919.

[7] S. Aljanabi and A. Chalechale, "Improving IoT Services Using a Hybrid Fog-Cloud Offloading," *IEEE Access*, vol. 9, pp. 13775–13788, 2021, doi: 10.1109/ACCESS.2021.3052458.

[8] A. Patel, M. Daftedar, M. Shalan, and M. W. El-Kharashi, "Embedded hypervisor xvisor: A comparative analysis," in *Proceedings - 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2015*, 2015, pp. 682–691. doi: 10.1109/PDP.2015.108.

[9] J. Almutairi and M. Aldossary, "A novel approach for IoT tasks offloading in edge-cloud environments," *Journal of Cloud Computing*, vol. 10, no. 1, Dec. 2021, doi: 10.1186/s13677-021-00243-9.

[10] V. Puri, I. Priyadarshini, R. Kumar, and L. C. Kim, "Blockchain meets IIoT: An architecture for privacy preservation and security in IIoT; Blockchain meets IIoT: An architecture for privacy preservation and security in IIoT," 2020.

[11] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4. Institute of Electrical and Electronics Engineers Inc., pp. 2292–2303, 2016. doi: 10.1109/ACCESS.2016.2566339.

[12] Q. Yang and H. Wang, "Privacy-Preserving Transactive Energy Management for IoT-Aided Smart Homes via Blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11463–11475, Jul. 2021, doi: 10.1109/JIOT.2021.3051323.

[13] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2. Institute of Electrical and Electronics Engineers Inc., pp. 1508–1532, Apr. 01, 2019. doi: 10.1109/COMST.2019.2894727.

[14] B. W. Nyamtiga, J. C. S. Sicato, S. Rathore, Y. Sung, and J. H. Park, "Blockchain-based secure storage management with edge computing for IoT," *Electronics (Switzerland)*, vol. 8, no. 8, Aug. 2019, doi: 10.3390/electronics8080828.

[15] B. Confais, A. Lebre, and B. Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS," in *Proceedings - 2017 IEEE 1st International Conference on Fog and Edge Computing, ICFEC 2017*, Aug. 2017, pp. 41–50. doi: 10.1109/ICFEC.2017.13.

[16] A. Mehbodniya, R. Neware, S. Vyas, M. R. Kumar, P. Ngulube, and S. Ray, "Blockchain and IPFS Integrated Framework in Bilevel Fog-Cloud Network for Security and Privacy of IoMT Devices," *Computational and Mathematical Methods in Medicine*, vol. 2021, 2021, doi: 10.1155/2021/7727685.
[17] P. B *et al.*, "Toward Blockchain for Edge-of-Things: A New Paradigm, Opportunities, and Future Directions," Apr. 2021, doi: 10.1109/IOTM.0001.2000191.
[18] M. A. Rahman, M. M. Rashid, M. Shamim Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-Based Cognitive Edge Framework for Sharing Economy Services in a Smart City," *IEEE Access*, vol. 7, pp. 18611–18621, 2019, doi: 10.1109/ACCESS.2019.2896065.
[19] P. Maymounkov and D. Mazì, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric." [Online]. Available: http://kademlia.scs.cs.nyu.edu
[20] I. Baumgart and S. Mies, "S/Kademlia: A practicable approach towards secure key-based routing," in *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, 2007, vol. 2. doi: 10.1109/ICPADS.2007.4447808.
[21] S. Nan and Z. Su, "Spark vs. Hadoop MapReduce."
[22] R. Hossen *et al.*, "BDPS: An Efficient Spark-Based Big Data Processing Scheme for Cloud Fog-IoT Orchestration," *Information (Switzerland)*, vol. 12, no. 12, Dec. 2021, doi: 10.3390/INFO12120517.
[23] C. H. Hong and B. Varghese, "Resource management in fog/Edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Computing Surveys*, vol. 52, no. 5, Sep. 2019, doi: 10.1145/3326066.
[24] A. Mijuskovic, A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga, "Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification," *Sensors*, vol. 21, no. 5. MDPI AG, pp. 1–23, Mar. 01, 2021. doi: 10.3390/s21051832.
[25] B. Amento, B. Balasubramanian, R. J. Hall, K. Joshi, G. Jung, and K. H. Purdy, "FocusStack: Orchestrating edge clouds using location-based focus of attention," in *Proceedings - 1st IEEE/ACM Symposium on Edge Computing, SEC 2016*, Dec. 2016, pp. 179–191. doi: 10.1109/SEC.2016.22.
[26] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "Blockchain-Based, Decentralized Access Control for IPFS," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1499–1506. doi: 10.1109/Cybermatics_2018.2018.00253.
[27] L. Xu, I. Markus, I. Subhod, and N. Nayab, "Blockchain-based access control for enterprise blockchain applications," in *International Journal of Network Management*, Sep. 2020, vol. 30, no. 5. doi: 10.1002/nem.2089.
[28] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, Jun. 2018, doi: 10.1109/ACCESS.2018.2851611.
[29] A. Tiwari and U. Batra, "IPFS enabled blockchain for smart cities," *International Journal of Information Technology (Singapore)*, vol. 13, no. 1, pp. 201–211, Feb. 2021, doi: 10.1007/s41870-020-00568-9.
[30] Z. Ning, L. Xiao, W. Liang, W. Shi, and K. C. Li, "On the exploitation of blockchain for distributed file storage," *Journal of Sensors*, vol. 2020, 2020, doi: 10.1155/2020/8861688.
[31] H. Honar Pajooh, M. A. Rashid, F. Alam, and S. Demidenko, "IoT Big Data provenance scheme using blockchain on Hadoop ecosystem," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00505-y.
[32] Q.-V. Pham *et al.*, "A Survey on Blockchain for Big Data: Approaches, Opportunities, and Future Directions."
[33] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, "When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues," *IEEE Access*, vol. 8, pp. 50574–50586, 2020, doi: 10.1109/ACCESS.2020.2979881.