Difuon Fog–Decentralized Federated Ephemeral Edge Computing Platform: *Market and Technology Opportunity*

Samuel M. Smith Ph.D.[1]

Original 2018/10/20 Version 0.1, 2018/11/30

Abstract— The coming tidal wave of pervasive devices both personal and Internet-of-Things (IoT) will not realize a fraction of their potential value without an equally pervasive infrastructure for data integration and analysis at the edge where the devices live. Currently data integration and analysis happens in the cloud with the associated higher latency and ingestion costs. We call computing infrastructure in the edge, fog computing. The fog is like the cloud only much more distributed. Because fog computing leverages computing resources much closer to the sources (producers, sensors) and sinks (consumers, actuators) of data, it has the potential for better scalability, higher performance and lower cost.

The distributed nature of IoT means that processing and analysis scales better if aggregated in a bottom up hierarchy from local to global. The data integration layer needs to be universal, not fragmented as it is now. It needs to leverage as much as possible the existing cloud based tooling for distributed applications but adapted to the unique characteristics of the fog.

Given the rampant fragmentation, stratification and siloization of existing IoT device level protocols, frameworks, and platforms, the only viable solution is an Internet Protocol (IP) centric approach that enables reliable, secure, decentralized integration across and through IoT silos and strata. This solution is a decentralized federated ephemeral edge computing framework, platform, and marketplace for data integration and analysis. This is the Difuon Fog. The Difuon Fog exploits and integrates several emerging technologies to provide a decentralized computing infrastructure at scale for personal data applications, Internet of Things (IoT), Industrial Internet of Things (IIoT), Internet of Tiny Things (IoTT).

Index Terms— *Edge, Fog, Computing, Decentralized, Federated, Platform, Ephemeral, Data Economy*

1. Introduction

Digital age technology is transforming economic and social activity in increasingly pervasive ways. One exponentially growing driver of this transformation is what may be called the *data economy*[3; 29; 94; 133; 138; 140]. The data economy is all about the monetization and capture of value from the data being extracted, transported, processed, refined, and stored by the increasingly pervasive digital devices and sensors employed in greater and greater portions of all forms of daily activity. These devices include not just mobile phones, pads, watches, and computers but devices used for digitization and streaming of content for entertainment and sharing, as well as ambient intelligent[48; 121] devices like voice controlled smart speakers, and media controllers as well as all the embedded, increasingly ubiquitous, sensors, actuators and other devices that comprise the internet of things (IoT). In addition to data generated by actual devices, there is growth in data generated by *virtual* devices, such as the instrumentation built into software applications, especially mobile and web applications, to monitor, track, and graph online activity. This pervasive digitization evokes the image of a virtual rain of data falling all around us, a *digital data rain*. Unlike the green digital rain that is the essence of the *virtual* sci-fi world in the Matrix movies[147], however, this *digital data rain* is a solution of measurements of the *real* world. Moreover, through collection, mixing, and distillation, the *data rain*, can be used to build virtual digitized views of relevant segments of the *real* world, that become, monetizable data products to help decision making, transactions, and/or improve profitability and affect the quality of life. This perspective poses the questions; who will capture the latent value in this *data rain* and how will that capture occur? Likewise, what nascent opportunities have arisen due to the exponential growth in demand for network and computing infrastructure to collect, mix, and distill the *data rain* and how best to take advantage of those opportunities?

In order to answer these questions it helps to understand the current state of computing infrastructure. Most processing of data happens in data-centers. These are essentially warehouses filled with computing hardware consisting of racks of compute servers together with storage, network connections, and reliable power. A staff of IT professionals manages and maintains this computing hardware. The hardware, software, and IT staff together comprise a computing infrastructure. The common term used to describe data-center based computing infrastructure is the *cloud*. Because data-centers are usually located far away from both the sources of data and the users of the applications, the metaphor of a distant *cloud* is an apt one for this infrastructure. There are two main types of *clouds*, *private* and *public [61]*. A *private cloud* is composed of compute servers in data-centers owned or controlled by an entity solely for its own use. Typically the servers in a *private cloud* are isolated behind a firewall in an internal network that limits access to external devices. Running a *private cloud* usually requires a dedicated IT staff to manage the associated servers. Smaller entities may rent rack space in someone else's data-center to house their privately owned servers. A

1. Senior Member IEEE, CTO Co-Founder Difuon.

larger entity may have its own dedicated data-centers. A *public cloud*, on the other hand, is composed of servers in data-centers owned and managed by an entity that rents the use of those servers to other entities. Typically the servers in a *public cloud* all belong to the same network, but each server may have its own firewall that limits access to itself. The *public cloud* host entity provides IT staff to manage and maintain the hardware as part of the rental fee. In order to be able to manage its rented servers the *public cloud* host IT staff has login access to the operating systems of those servers. There are also *hybrid* clouds that have different compositions of private and public components and control over those components[56; 58].

Traditionally, public clouds were used in concert with private clouds to add elasticity to the private cloud in order to scale capacity up and down in response to dynamic variations in load. This reduced the need for expensive overcapacity to handle dynamic peak processing loads. Thus, despite being more expensive per unit of computation, public clouds, on balance, are often more economical overall by avoiding the expense of unused capacity. Increasingly moreover, because of the convenience of not having to maintain dedicated computation infrastructure including the associated IT staff, especially for smaller entities, public clouds are not just being used to augment private clouds but are instead completely replacing private clouds. Furthermore the operational advantages of expensing the rental of infrastructure as a service (IaaS) versus maintaining infrastructure as a capital asset is attractive to startups. Once an entity replaces its private cloud infrastructure with public its difficult to ever go back. One factor stopping a wholesale migration to public cloud from private cloud are concerns about security and privacy of the public cloud. In one report 66.5% of companies with private cloud are concerned about public cloud security threats to sensitive data and 56.9 percent were concerned about third party account compromise[17]. A related problem is the competitive threat posed by public cloud providers, such as, Microsoft, that also market software applications. Soon after Microsoft acquired the cloud based source code repository service, Github, over 100,000 plus users switched to competing services citing, as one of many, concerns over the confidentiality of their source code in that hands of a potential competitor [56; 72].

Historically, in terms of market share for application workloads, private clouds had a much bigger share relative to public clouds. This has been rapidly declining, one estimate is that in 2016 the ratio of private to public cloud application workloads was at 60.9/39.1%. Whereas by 2018 this ratio will have flipped, such that the majority 53.8% of application workloads will be in public clouds [17; 24].

What makes the trend of increasing public cloud dominance so concerning is that a handful of Fortune 100 providers control most of public cloud infrastructure and that portion is increasing relative to their competition[73]. In terms of revenue, in 2016 Amazon had a 40 percent share, with Microsoft, Google, and IBM together making up a 23 percent share for a total of 63%. That total share was an increase of 5 percent from 2015 [67]. It is predicted that in 2018 this dominance will grow such that the combined Amazon, Microsoft, and Google share of the public cloud market will be 76 percent and keep growing to 80 percent by 2020 [74]. Thus while the public cloud is capturing more and more of the overall compute infrastructure market share, the top four public cloud providers, with Amazon much more than the other three combined, are capturing an increasingly larger relative share of the public cloud market. This trend towards a technical monopoly of compute infrastructure as the data economy becomes an increasingly valuable segment of all economic activity is particularly worrisome.

From the broader perspective of answering the question of who will capture the most value from the data economy, the five largest companies by market cap are Apple, Amazon, Microsoft, Alphabet(Google), and Facebook[118]. In May of of 2018 the combined market cap of these five was over 3.7 trillion USD[126] and is increasing. Apple and Amazon for the first time each surpassed the one trillion mark[118]. Notable is that three of the five are also the three largest public cloud providers and all derive a significant share of their value from the monetization of their large private clouds.

The most important insight, however, to understand in terms of recognizing opportunities is not the landscape of current computing infrastructure as controlled by a handful of cloud providers, but to understand what the landscape of future computing infrastructure could become. As mentioned above, the data economy is increasingly driven by data coming from internet connected devices. One estimate gives the number of internet connected devices at 11 billion in 2016, tripling to 30 billion by 2020, and then almost tripling again to 80 billion by 2025 [82; 139; 140]. The amount of associated digital data is expected to increase from 4.4 zettabytes[2] in 2013, to 180 zettabytes in 2025 [82; 139]. By in large the source of this data is not from servers in the cloud but from devices outside the cloud. The portion of the internet that consists of devices that are not in data-centers is called the *edge [78]*. Thus most sources of data reside in the *edge* and are called *edge* devices. The current situation is that in order to process the data generated in the edge, the data must be first sent to remote data-centers, ie, the cloud. This adds cost, latency, rigidity, and complexity [80; 104]. These problems are compounded when the results of that processing are needed back in the edge. Much of the associated data processing is by nature aggregation, integration, and analysis, which in many cases can be performed in a bottom up manner. Consequently much of the processing does not need to be performed in the cloud. Unfortunately however, there is little or no current computing infrastructure in the edge so the data is sent to the cloud by default for processing. Herein lies the opportunity! In order to handle the coming exponential growth in the amount of data processing, which some liken to a tsunami of data [85], there will needs be an exponential growth in the capacity of computing infrastructure. Moreover most of the new computing infrastructure could better reside in the edge not the cloud [4; 80; 113; 133]. Because most data will be created in the edge, compute on that data will likely be driven to the edge as well[146]. Employing the same *cloud* metaphor, computing infrastructure located in the *edge* is often called the *fog* because unlike servers in far-away cloud, *fog* computing servers are nearby the sources (producers, sensors, personal devices) and sinks (consumers, actuators, applications) of data [6; 12; 14; 45; 46; 89; 146].

2. 1 zettabyte = 1 trillion gigabytes

Indeed, returning to our original metaphor, the current *data rain* is but a sprinkle, but is soon to become a torrential *downpour* as tens of of billions of new devices are connected to the internet in the next five years. Collecting, mixing, and distilling value from this torrential downpour will require new computing infrastructure that is best located in the *fog* not the *cloud*.

The big public cloud providers are providing some support for processing data in the fog. They all offer hybrid public/private fog computing platforms. Amazon has AWS GreenGrass which enables installation of Amazon software on private compute hardware for running local instances of AWS Lambda [52]. Microsoft has Azure IoT Edge which is more full featured and allows local Azure instances of Function, analytics, and machine learning[10]. Google has Cloud IoT Edge which allows private edge devices to run Google software that allows some local processing [23]. IBM has a beta of software products for its IoT Edge Analytics initiative [38; 49]. These offerings do not yet provide true public fog computing infrastructure but could be seen as movement in that direction. Until there is a viable competitive threat, there is very little incentive for the public cloud providers to invest in the hardware needed to provide a public fog. The up front capital cost and lead time for building large data-centers is daunting. Building a new public fog computing infrastructure, if it follows the cloud model, will be expensive and time consuming. This poses the questions: will the future of public fog computing be controlled by the existing public cloud providers? Are there opportunities to disrupt the public computation infrastructure market as it moves to the fog? The answer is to the latter question is yes. The next section motivates the core elements of our plan for capturing a meaningful segment of the emerging fog computing market.

2. The Future of Public Computing Infrastructure

Practically speaking, public fog computing infrastructure does not yet exist, but is inherently potentially much better suited than public cloud computing for processing the data from an exponentially growing number of edge devices[133]. This presents both a huge problem and huge opportunity, that is, how to best build the future public fog computing infrastructure and capture value from it?

When referring to the public compute infrastructure market, the term *centralized* (*decentralized*) has two relevant meanings [141]. One meaning is political, as in the degree to which the ownership and control the compute elements are spread across multiple entities. The other meaning is architectural, as in the degree to which the compute elements and functions are spread across multiple computer hosts and physical sites. To avoid confusion, this paper uses *centralized* (*decentralized*) for its former political meaning and uses the term *distributed* (*non-distributed*) for the latter architectural meaning. Because compute infrastructure that is more distributed tends to be less vulnerable to failure and more responsive to dynamic load it tends to have much higher availability. As a result much of public compute infrastructure is highly distributed, albeit more complex. One attraction of public cloud is that its providers reduce the time and effort for their customers to build, manage, and operate distributed compute systems. Computing infrastructure under more centralized control, however, tends over time, to be more expensive and less innovative as the providers are able to extract more value in the absence of competition. Thus, despite the initial cost savings due to the scale effects of large centralized clouds, the ability to control pricing independent of costs eventually dominates. Furthermore, public compute infrastructure under centralized control may expose the customers and owners of the associated data to higher privacy and security risks. Centralized control can be a common mode failure source for the distributed system as a whole thereby making it not just less private and secure but also less available[59; 65; 116].

The goal is to find a way to minimize the problems of centralized control without giving up its scale benefits. The best way to achieve the cost savings at scale without having to build at scale is to leverage spare capacity, when available, and commoditized components otherwise. The key enabler is a platform that makes available at zero or low cost unused or under used capacity. The base commodities needed for a public fog computing infrastructure are space, energy, and bandwidth to house, power, cool, and connect the compute servers. Making the servers work as part of a computing infrastructure also requires software to manage the servers automatically as well as to run a marketplace for the associated products and services. The solution we are building is a public decentralized distributed fog computing infrastructure that leverages spare capacity, commoditized components, and leading edge software to build out at scale, at an otherwise unachievably high rate and low cost. Our approach takes advantage of an precedented convergence of technologies and market trends which include distributed consensus, blockchain, serverless computing, disposable servers, automated reasoning, zero trust computing, and tokenomics driven network effects, to name a few. The pieces of the approach are outlined below.

2.1. Spare Capacity

The base commodities needed for a public fog computing infrastructure are space, energy, and bandwidth to house, power, and connect the compute servers. In the cloud these are provided in specially constructed data-centers with dedicated custom built electrical utility power substations and network connections. Building a new data-center may take years and may require large capital outlays not to mention the effort to manage geographic and political complications. Fog computing infrastructure needs to be close to where the data is created, in other words, located nearly everywhere. One approach some are taking is to build micro-data-centers [69], although viable, that approach still requires purchasing space, power, and bandwidth for those micro-data-centers. Our approach is to use spare space, energy, and bandwidth that is already available in residential, and commercial buildings. Of these the limiting constraint is bandwidth. Fortunately there is an increasing availability of unused bandwidth especially in residences with broadband connections. Bandwidth is often measured in GB (gigabytes) of data transfer. Total available data transfer can be estimated by multiplying the internet connection speed in Gbps by a time interval in seconds, such as, seconds per month. Average spare bandwidth in GB is then just the average available bandwidth minus the average used bandwidth. Estimates of average and median internet connection speed vary widely and often combine mobile and fixed

connections, the trend, however, is clear, average connection speed is increasing year over year [21; 42; 42; 68; 91; 91; 148]. One report estimated that in 2018 47 million (37%) of US households will have at least a 100 Mbps download connection speed [90]. This seems to be an underestimate as the continually updated Speedtest Global Index showed that for the month of August 2018, that the United States, the number six ranked country, had an average fixed connection speed of 100.07 Mbps download and 33.58 Mbps upload [124]. The same index shows that for the same month, the average global fixed connection speeds as 47.83 Mbps download and 23.02 Mbps upload [124]. Being conservative, an average connection speed of only 50 Mbps provides a maximum total available data transfer of 16.2 TeraBytes per month = 50 Mbps * 2,592,000 seconds per month. Practically speaking 100% bandwidth utilization is not possible. The average broadband residence in the US used 190 GB per month in 2016 [60]. The average per user was 106 GB per month [90]. But even with fractional utilization the available spare capacity is significant. Increasing connection speeds above a certain level does not significantly increase usage as about 80% of residential usage is for streaming video and the number of videos downloaded is limited by the number of viewable hours in a day [60; 90]. Consequently, the growth in faster connection speeds is resulting in lower latency, faster downloads, and more spare bandwidth. By way of comparison, typical data transfer allowances for cloud virtual machine computers are typically from one to a few TB per month. This indicates that a 50 Mbps broadband connection should have sufficient spare bandwidth to run one or more fog compute servers. Some broadband providers have data caps on total GB transferred with an extra fee to increase the data cap. Nonetheless, the average cost for available bandwidth will still below even if no longer at zero cost.

2.2. Disposable Servers

It is notable that spare compute capacity was not included in the prior section. Many have stated the assumption that personal laptops, desktops, and mobile devices have ample idle capacity and therefore could be used for the public compute servers [1; 28; 132]. We believe that assumption to be largely false, primarily, because most users will not put up with the inconvenience, security vulnerabilities, and privacy violations that result from running external compute jobs on their personal hardware. These concerns severely limit the actual obtainable capacity of personal devices for public compute services. This limitation means that merely relying on the spare capacity of personal compute devices is not a scalable solution. A more viable approach is to use dedicated compute servers instead. These need to be inexpensive and easy to deploy by non-tech savvy building occupants. Although the cost of commodity servers is largely a function of volume. Sometimes ancillary markets drive the cost of a given class of server to near its minimum achievable high volume cost, even a low individual purchase volumes. When that happens the class of compute server is at a so called sweet spot for price for performance. When that price point is low enough to be incidental to the value of the services provided the compute server hardware is essentially disposable. In other words, a *disposable server's* hardware is so inexpensive that it is incidental to the cost of its provided services.

Fortunately there is a family of computer that can be beneficially employed as essentially disposable compute servers. This family is at the sweet spot of computing in terms of an optimal combination of cost, memory, speed, storage, size, and energy. They are ARM based computers that run Linux or some form of Nix like BSD. They also support an open source trusted execution environment [88]. Several manufacturers make these *Raspberry Pi* like devices [87; 100; 102]. They have credit card sized boards with similar layout such that the same enclosures can be used for nearly any member of the family. They are very low power (< 15 Watt) ARM devices and often do not need fans for cooling. One available configuration provides 4 cores at 1.5 GHz, 4 GB of RAM, GPU, and 1 Gbps Ethernet at a quantity one pricing of under $50 USD [102]. When fully outfitted with 128 GB of flash disk space, enclosure, and power supply the price is still under $100 USD. A comparable cloud virtual machine costs almost that much each month. This means that the cost of the hardware may be amortized in as little as one or two months. The usable life span of the hardware is at least two to three years, thus making its cost incidental to the total lifetime value of the compute services it may provide. These *disposable servers* combined with the spare space, energy, and bandwidth from existing broadband capable buildings may provide a near zero cost basis for a public fog computing infrastructure.

2.3. Ephemeral Computing

Ephemeral means short lived, temporary. Thus *ephemeral computing* means a computing environment which is short-lived. An *ephemeral compute* environment is set up on demand for a given computing task and then torn down once the task is complete [27]. *Ephemeral computing* provides on-demand availability from computing resources that may be too volatile, too diffuse, or too expensive to be continuously employed. One particular type of computing environment is a virtual server. Typically, a consumer of compute will rent a virtual server and then configure it with software services. Because the server and its services are long running they are not ephemeral. The customer is responsible for maintaining and updating the server's software. Recently the major cloud providers started offering a new class of service called *serverless* or *functions as a service (FaaS)[115]*. This new class of service is a type of *ephemeral* cloud computing environment For example, Amazon offers AWS Lambda, Google offers Cloud Functions, Microsoft offers Azure Functions and IBM offers OpenWhisk. With serverless there is no persistent virtual server that the consumer must maintain. Instead the services are run on demand by the serverless environment [9]. The compute infrastructure provider is responsible for maintaining the underlying server infrastructure. This frees the consumer from the having to manage virtual server infrastructure. The consumer merely specifies the computing requirements to run the services in terms of triggering events, software libraries, memory, computation, and network. Typically serverless APIs support running individual functions that can be composed into full fledged services or micro-services [83]. The functions are executed based on events that are specified as part of the serverless configuration parameters. The functions are run upon event occurrence. This setup provides dynamically available (on-demand) resources to run the functions as more events occur. Another approach to ephemeral computing is to use containers that are created on demand to in response to load [19; 37].

These well known applications of ephemeral computing can better optimize the cost of computing resources and make them more convenient to use.

Because serverless divides the functionality of services into small chunks that may be run as individual functions or at most micro-services, the needed computing resources per serverless instance are much smaller. Consequently the serverless approach is inherently more compatible with the limited computing resources of disposable servers.

2.4. Federated Computing

Federated computing is where cooperating distributed servers together perform computing tasks [36]. *Federated computing* coordinates computing resources from disparate locations to provide a shared computational resource. Historically, federation has only been used on large computing tasks such as high performance, grid, or big data computing. The approach of building more complex services from a set of serverless functions is a new type of *federated computing*. Typically *federated computing* has only been implemented with a centralized coordinator. Federating decentralized fog computing resources can be problematic due to the diffuse control of the associated computing resources and the extremely sparse nature of any single providers's fog resources. This may require federating fog resources among disparate providers; That is the challenge. Locality concentrated fog computing resources, however, may provide the best price for performance for edge computing; That is the opportunity. Fortunately, with an appropriate decentralized architecture, federation techniques can be applied to combine a group of small disposable servers running a serverless framework to perform a bigger computing task, that is, a full fledged cloud-like service, but in the fog.

2.5. Decentralized Computing

Decentralized computing means that the computing resources are under the control of multiple parties. This requires some mechanism that enables these parties to cooperate and coordinate the use of their resources in a single computing infrastructure. As mentioned above, with such a mechanism, a decentralized fog computing infrastructure would make available new inexpensive computing resources that would otherwise be under-utilized due to their diffuse and volatile nature. Moreover when equipped with federated ephemeral computing software, multiple fog devices could be configured to provide the equivalent of a much more powerful cloud computing server. Thus decentralized federated ephemeral computing may have the highest potential for the optimized utilization of fog computing resources.

2.5.1. Distributed Consensus

The most promising method for decentralized coordination of resources under the control of multiple parties is a *distributed consensus* algorithm. Distributed consensus is the process by which a pool of computing nodes come to agreement about a sequence of events. This sequence might be transactions in a ledger, or entries in a database, or the order of operations in a state machine. There are three main classes of distributed consensus algorithms, these are: *Proof of Work* (POW) [98; 112; 131], *Proof of Stake* (POS) [97; 145], and *Byzantine Agreement* (BA) [7; 15; 20; 22; 32; 33; 40]. There are many variants of the three main classes of BFT algorithms. What they all share is a feature called *Byzantine Fault Tolerance* (BFT) [15]. Simply, BFT means that that despite faulty behavior that includes malicious acts by a minority of the members, the pool as a whole will still come to consensus.

By way of comparison, the basic concept of PoW is that computing nodes compete to solve a cryptographic puzzle. The winner is the first one to broadcast proof of the solution. The winner becomes the author of the next block of transactions. Variants use different types of cryptographic puzzles or work. Some variants use actual or useful computation work. Often the type of work is best suited for different types of computing hardware such as ASIC, FPGA, or GPU. PoW is a simple algorithm but its drawbacks are high latency and low throughput.

In contrast, the basic concept of PoS is that some verifiable staking function biases the selection of the node that is the author of the next block of transactions. One way to think of PoS is as an asymmetric proof of work where a larger stake reduces the amount of work. Those with larger stakes win faster and more often. Thus PoS has lower latency and higher throughput than PoW, albeit with more complexity to manage the staking. A popular variant of PoS is Delegated Proof of Stake (DPoS) where participants vote their stake to elect the author(s) of the next block of transactions [35]. Another well known variant is Federated Proof of Stake (FPoS) where participants select a subset of nodes to come to consensus on their specific transactions [32]. Further variants are sharded or hierarchical PoS where the nodes are broken up into sub-groups (shards) that form consensus on a subset of transactions and then merge the subsets [142].

Predating both PoW and PoS are the BA algorithms. In BA algorithms the number nodes in the pool is fixed. The first popular usable BA algorithm was called Practical Byzantine Fault Tolerance (PBFT) [20]. In PBFT, the nodes in the pool elect a leader. The leader then authors the next block of transactions and submits it to the pool for validation using a three phase commit. The main advantage of BA is that compared to both PoW and PoS, the latency is much lower and the throughput is very much higher. The disadvantage of BA is complexity and the number of pool members much be kept low. Several variants of PBFT have been developed. Notable is Redundant Byzantine Fault Tolerance (RBFT) which is more resilient [7]. Others include Fast Byzantine Consensus (FBC) [77]. Others use some type of directed acyclic graph (DAG) to that makes the consensus leaderless and more asynchronous [71; 114]. Finally a hybrid variant of BA and PoS, which we call Randomized Byzantine Agreement, uses a verifiable random function to come to consensus [43; 99]. Variants include biasing the random function by the stake to select the author(s) of the next block or other similar ideas [8; 50; 103; 135]. In the last two years there has been an explosion in the number and type of variants of BFT distributed consensus algorithms [26; 149]. It is beyond the scope of this paper to mention all of them or to describe any of these in detail.

The design of a BFT implementation requires making tradeoffs between the features of the different algorithm classes. The salient features that usually one must trade-off are security, scalability, and governance. There are four primary design elements that a BFT distributed consensus implementation must specify and provide, these are:

- The governance policy for selecting pool members
- The distributed consensus algorithm(s)
- The data structure holding the events managed by the pool.
- The reward mechanism for incentivizing participation of the pool members.

The resultant distributed consensus system is then used to provide transactions that support a given application or applications. A system design task is to compose the best combination of these four elements to best match the transactions for the application at hand. For example one very common data structure used in distributed consensus systems is a ledger. Because the pool members are distributed and each has a copy of the ledger it becomes a distributed ledger. The ledger is often implemented as a blockchain to enable integrity verification. It is notable that the term *blockchain* is now commonly used to mean more than just a type of data structure [129]. Its more general meaning refers to a host of technologies that include crypto-currencies, distributed ledgers, distributed consensus, modern crypto, and decentralized identity to name a few. Solving a business problem using *blockchain* in its more general sense means selecting and integrating a specific mix of the aforementioned technologies. The particular mix is typically targeted to the application that solves the business problem.

The most popular BFT distributed consensus implementations are PoW distributed ledgers. These includes BitCoin and Ethereum [13; 41]. The attractive feature of PoW is that the pool governance policy is included as part of the PoW consensus determination. This allows for open public participation of an indeterminate number of pool members. As previously mentioned, the main limitation of PoW is high latency and low throughput. PoW by itself is not scalable to thousands of transactions per second. Consequently much of the recent innovation in bitcoin technology has been to find ways to address the latency and throughput limitations of PoW. One solution is to use a PoS system, which usually has higher throughput and lower latency than PoW but may require more careful management and governance in determining the stake. As previously mentioned, PoS can be thought of as an asymmetric PoW where the size of the stake determines how much work must be done. In contrast, BA algorithms have the fastest throughput and lowest latency of any of the three algorithm types. The main drawbacks of BA are its complexity and the total number of pool members is limited.

2.5.2. Hybrid Systems

The approach that we believe has the most promise is to use a hybrid system. In a hybrid system, multiple distributed consensus mechanisms and pools given a more optimal combination of security, scalability, and governance. For example, one way to achieve both high security and high throughput is to use a PoW ledger as a periodic anchor to transactions that happen on another faster non-PoW auxiliary distributed consensus pool. This is often called a side-chain [2]. Another approach is to use a different consensus mechanism for pool governance than for transaction ordering. In the fog computing application, compute job allocation must be completed with low latency and at high throughput. Consequently, the only viable choice for the auxiliary consensus algorithm is a type of BA algorithm. Because pool membership governance is not built into BA algorithms, using BA requires implementing a governance policy or algorithm to manage pool membership. Because pool membership changes are a relatively infrequent event, however, a slower consensus algorithm such as PoW/PoS can be used to automate pool membership governance. Periodically anchoring the BA pool either to a much larger or to multiple PoW/PoS pools provides increased security while still enabling very high transaction rates. We originated a form of this called ChainMail [108]. A distribution algorithm such as network distance or some type of distributed hash table (DHT) to assign transactions to different pools makes it scalable. We believe that in general this is the most viable approach for decentralized fog computing infrastructure.

We have deep experience with BA algorithms. One is *Plenum*, a modern BA implementation based on the RBFT algorithm hosted by the HyperLedger Indy project [95]. We have also originated a next generation BA algorithm called TOBA (Timeliness Ordered Byzantine Agreement). TOBA is a leaderless asynchronous hybrid algorithm that combines elements of a conventional multi-phase commit with a directed acyclic graph (DAG) to make it both more resilient and more scalable. One advantage of TOBA is that the core algorithm lends itself to an automated reasoning algorithm that aggregates proof of useful work, delegated proof of stake, and expected reliability to select pool members (governance).

2.5.3. Design Issues

A careful examination of the scalability proposals for PoW blockchains such as Ethereum and Bitcoin reveal that scalability is not possible with a single distributed ledger that tracks all transactions as a single asset class. Proposed fixes include some combination of sharding (multiple asset classes with receipts between shards) or side chains [2; 142]. A more scalable approach is to use triple signed receipts on transactions amongst mutually disjoint sets of parties to the transaction [53; 54; 84; 101; 125; 136; 143]. The receipts provide non-repudiable proof of the agreement at each stage of a transaction. The consensus pool acts a trusted third party to sign the receipts that clear transactions within a given asset class or between two or more asset classes. These completed transactions can then anchored to a slower cryptocurrency based blockchain such as BitCoin and Ethereum.

A decentralized identity system provides the core mechanism for allowing interaction between different pools and different asset classes. A performant asynchronous peer-to-peer communications protocol that uses decentralized identity makes the approach scalable. Both TOBA and the first implementation of Plenum use the RAET (Reliable Asynchronous Event Transport) protocol [107]. It is based on NaCL (LibSodium) and CurveCP which are recognized as the state of the art open source ECC

crypto libraries for key exchange, signing and encryption. RAET is an inherently scalable UDP/IP based protocol that uses best practices for secure reliable communications. We originally developed the open source RAET protocol for highly distributed cloud infrastructure automation, so it is well suited for fog infrastructure automation. One of the primary concerns of a fog computing infrastructure is the management and control of the computation and data on remote edge devices. Management of these devices requires a scalable reliable secure communications protocol. RAET is that protocol.

Building distributed applications requires interoperable communication protocols to connect software components on disparate servers. While there are many communication protocols that may be employed to connect software components, the most universal approach is to use IP (Internet Protocol) based protocols. Indeed, one of the biggest challenges in fog computing for IoT applications is the fragmentation, stratification and siloization of existing IoT devices due to the rampant use of non-IP based protocols. Historically, the limited computing resources of embedded IoT devices motivated the used of resource constrained protocols, but every year the frontier of applications that are too constrained for IP protocols is shrinking. The CoAP project is an example of lightweight IP implementations that are friendly for constrained resource networks [25]. Thus non-IP networks are now generally only desirable as legacy point optimizations, not universal solutions. The only viable universal solution for a public fog computing infrastructure is an Internet Protocol (IP) centric approach that enables reliable, secure, decentralized integration across and through legacy IoT silos and strata. Consequently Difuon is IP based. Access to/from other data sources and sinks that are not IP based may be provided via gateways. Moreover Difuon's IP centric nature makes it very convenient to leverage a multitude of cloud based software tools, protocols, databases, frameworks, and platforms.

Many blockchain applications involve some form of smart contract, that is, rules that execute in a trusted but automated way to manage a given contractual relationship. An important consideration in this contract management is that the rules be verifiable. We have developed a flow based programming and decision-making framework called Ioflo that uses a model based approach to program synthesis and analyses that allows for easier verification [105; 106]. This was originally developed for autonomous vehicle systems. Ioflo is asynchronous and thereby provides a framework for building scalable backends for data processing of transactions. Ioflo is a type of flow-based programming framework with many unique features that make it applicable to highly distributed asynchronous reactive systems [18; 31; 44; 57; 63; 64; 70; 92; 127; 128]. Difuon repurposes it to provide a fog centric serverless computing platform. The current "serverless" frameworks are cloud centric and largely proprietary. Because it is a flow-based programming framework it is uniquely suitable for IOT systems. Ioflo itself is mature but the repurposing as a dynamically allocated functional service is new.

Important to any blockchain application is privacy preservation. Typical fully public blockchains are problematic. The path forward is to use public information as a trust anchor but the vast majority of information is exchanged in pairwise transactions that are off-chain. We are innovators in this approach. Decentralized autonomic data (DAD) items based on DIDs (Decentralized Identifiers) enable streaming data in a scalable but secure manner [34; 110; 117]. Using portable W3C Decentralized Identifiers (DIDs) provides the foundation for an open decentralized trust system. This enables a portable decentralized attribute based identity for every entity and every thing including every item of data. Decentralized Autonomic Data item (DADi) chains build with DIDs provide self-contained end-to-end data provenance of each data flow.

2.6. Diffuse Trust Computing

Federation requires that disparate entities cooperate to share resources. Sharing resources raises the risk that data or assets may be compromised because the of the inability of one entity to trust the actions of another. The conventional approach to IP security is often described as defense-in-depth where each nested layer forms a secure perimeter that restricts access to higher levels of trust. These perimeters typically consist of some form of host:port based firewall. The outermost layer faces the "wild-wild-west" of the open internet where every interaction may be hostile. The next layer may be in a DMZ and/or behind a firewall. The following layer might be a trusted service running on a private network. VPNs provide a way for an external remote host to tunnel through the outer layers to access a trusted space. What a perimeter model assumes is that network locality can be trusted. Internet hosts and services running in an inner layer are allowed to share information with each other because they are in a trusted “safe” space. This conventional approach fails for many reasons. Fundamentally it fails because it does not account for the fact that hostile users exist everywhere not just on the open internet. Any mechanism that allows for lower privileged users to access a network layer provides an attack vector for that user to elevate their privileges thereby gaining access to everything in that layer. Hence, network locality is not security. Moreover, perimeter security is static and does not respond to dynamically evolving exploits.

The solution is to use a *diffuse trust perimeter-less security model*. This is an extension of an earlier approach known as a *zero-trust security model*. A simple way of explaining the *zero-trust* security model is the mantra, *never trust, always verify*. The paradigm of *Zero-Trust Networking* was first popularized in 2013 by the a NIST report [47]. More recently the principles have received much broader attention [51; 144]. Calling it *zero-trust* or trust-less is a misnomer. There is still trust. It is just applied in such a way that security is enhanced. We have extended this model to accommodate computing resources under decentralized control [109; 110; 117]. We call this a *diffuse trust perimeter-less security model* because the security policy is maintained with decentralized resources that diffuse the trust in such a way that security is even more greatly enhanced than merely zero-trust. A computing infrastructure that is built from the ground up using a *diffuse trust perimeter-less security model*, We call *diffuse-trust computing*. We believe this is the only truly viable approach for a public fog computing infrastructure. To our knowledge, Difuon is the first decentralized public fog computing infrastructure based on *diffuse trust computing* principles and may be the first public computing infrastructure of any kind based on these principles.

The assumptions and principles of *diffuse trust perimeter-less security* are: (see [109; 117])

- The network is always hostile both internally and externally; locality is not trustworthy.
- By default, inter-host communication must be end-to-end signed/encrypted and data must be stored signed/encrypted using best practices cryptography; Data is signed/encrypted at motion and at rest.
- By default, every network interaction or data flow must be authenticated and authorized using best practices cryptography.
- Policies for authentication and authorization must be dynamically modified based on behavior (reputation).
- Policies must be governed by distributed consensus.
- By default, each data flow including all transformations must be end-to-end provenanced using decentralized identifiers (DIDs) and hence decentralized autonomic data (DAD) items.

The first step to implementing diffuse trust computing principles is the recognition that any IP network is always hostile. This means all servers must be hardened using best practices for ports and services. Using end-to-end encryption and storage prevents exploits from anyone that merely has access to the network or the data storage device. By authenticating and authorizing every network interaction or data flow, security becomes granular. A successful exploit of one interaction does not bleed into any other. Compromising one data flow does not compromise any other. Escalation opportunities are minimized. Many security exploits are discovered through repeated probes and experiments to find bugs, buffer overflows, or weaknesses in network protocols or software implementations. Dynamic policy modification that uses a reputation AI to first profile and detect anomalous behavior and then restrict the authorization of that user prevents discovery. This adds time as a defense. Extending the model to use distributed consensus to govern the policy management including authentication and authorization enhances the security over using a centralized policy manager. Distributed consensus diffuses the trust for any policy decision to a group of hosts. In order to defeat the policy, an attacker must exploit some majority of the hosts. This makes exploits exponentially more difficult. Moreover. distributed consensus also allows for the use of decentralized computing resources for hosting the policy management pool and decentralized governance of the hosts. The final principle further extends the decentralized trust model to allow governance that leverages a decentralized web of trust based on the emerging W3C open Decentralized Identifier (DID) standard and Decentralized Autonomic Data (DAD) items built with DIDs [34; 110; 117]. These provide provenanced data flows that are compatible with big-data, data-flow, reactive, and flow based programming approaches needed for distributed data analysis tasks [18; 44; 55; 57; 63; 64; 70; 79; 92; 105; 106; 110; 117]. This approach enables truly decentralized governance models for distributed applications. One can combine the third and sixth caveats from above into a simpler summary caveat as follows:

- By default, all data flows are end-to-end provenanced/signed/encrypted at motion and at rest using DIDs and DADs.

The diffuse trust security model by itself addresses the security concerns of a decentralized public fog computing infrastructure. The other two main concerns are privacy and agency. We call these the SPA (*Security, Privacy, Agency*) design principles.

- Security
- Privacy
- Agency

Security, privacy, and agency are all extremely important to a decentralized public computing infrastructure. Many implementations only attempt to address one or two of the three. Difuon is designed to provide reasonable levels of all three. As previously mentioned distributed consensus helps improve security in the open.

Privacy is about minimizing the disclosure and leakage of of information to uninvolved third parties that may exploit it harmfully. For example, a fog computing transaction is typically between two different parties, the compute provider and the user of that compute. In order to consummate a transaction, the parties may need to share information about each other before proceeding with the transaction. In addition, the transaction itself may involve the exchange of information. This is obviously the case for fog computing transactions. A modular approach to privacy allows the participants to exchange on-chain (public distributed consensus pool) as little as possible and exchange the bulk off-chain (private peer-to-peer). A detailed exposition of privacy concerns and methods is beyond the scope of this paper but the core principles may be found in the provided references [108; 111; 122; 123]. Although the citations include white-papers from the Sovrin Foundation, the Difuon Fog is not an implementation of Sovrin but is based on similar principles such as privacy by design, minimum disclosure, and pair-wise unique identifiers for each relationship. Like Sovrin and others in the Decentralized Identity Foundation (DIF), the Difuon Fog does use the open portable DID standard [34; 62].

What is worthy of further explication is the degree of privacy preserved by the system. One way to classify privacy preservation is as being either *strong* or *weak*. *Strong* privacy preservation is achieved when any third party correlator can not practically extract any effective correlation between disparate identifiers used by an entity across multiple transactions. Theoretically one could contemplate *strong* privacy preservation for entities that follow a policy of a one time interaction with any other entity and the set of attributes used in all their interactions is mutually disjoint with respect to the interacting entities. However it would be difficult for an entity practicing such behavior to find other entities willing to interact with it. Thus making such a policy impractical. Generally speaking, any entity that regularly interacts online with other entities will eventually be correlatable because many third party correlators have practically unlimited storage, compute power, network access and time to make correlations. Consequently, we believe *strong* privacy preservation is not practically achievable especially for entities that

gain value from regular online interaction. In contrast, *weak* privacy preservation is achieved when the cost of correlating disparate identifiers is more than the value exploitable from that correlation. This minimizes the economic incentive for third party correlators to engage in correlation. *Weak* privacy preservation, therefore, is an economic problem and is practically achievable for many applications. Nation state correlators may not be inhibited by economic considerations for a given entity but may be inhibited if the number of entities is large enough. Thus weak privacy preservation is at best achievable on average with respect to nation state correlators and may be achievable individually for correlators with much more limited resources. One of the contributing factors is that both the value to the correlator and the harm to the correlatee tends to decrease the longer the time elapsed after a correlatable interaction has occurred. Thus weak privacy preserving systems can have significant temporal components.

Agency deals with the control and governance of the associated data and transactions. In fog computing, centralized governance is problematic. Distributed consensus algorithms provide a practical way to implement decentralized governance. Indeed as the foregoing discussion has shown, using distributed consensus to manage security, privacy, and agency provides a beneficial synergy.

2.7. Notional Architecture

This section provides an overview of basic architecture of the proposed fog computing infrastructure. It includes brief descriptions of the various components and the role they play. Companion white-papers will explain in more depth the associated protocols and algorithms. The following diagram provides a graphical representation of the various participating components in a full blown public fog computing infrastructure.

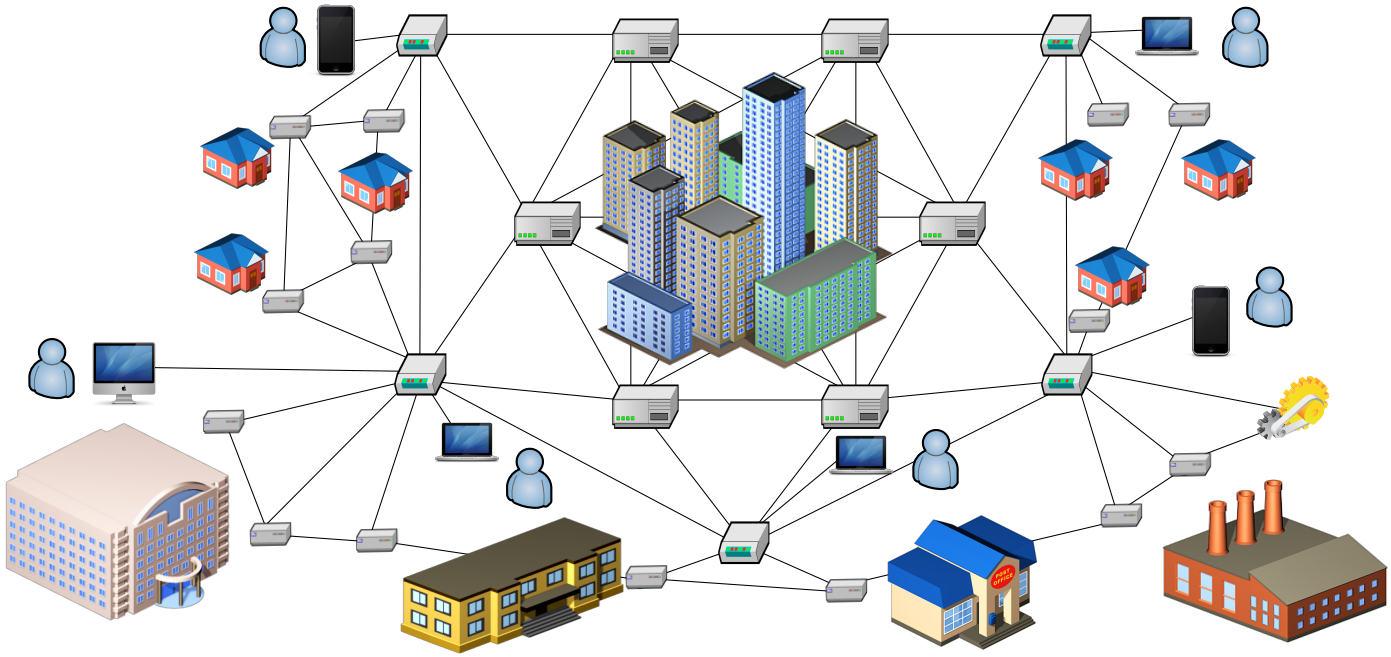

Fig. 1. Public Fog Computing Infrastructure

2.7.1. Distributed Consensus Pool

The core component is a Byzantine Agreement (BA) distributed consensus pool. The main feature of the BA pool is that it provides rapid distributed consensus that includes the ability to support nano-transactions. By nano-transactions we mean transactions that settle very quickly and for very small amounts. This enables flexibility in the composition and renumeration of federated server groups that provide ephemeral micro-services and functions as a service (FaaS). Primarily these transactions will be tracked on a distributed ledger of account balances maintained by the BA pool. This provides the basis for payment. In addition to the ledger of accounts the BA pool will maintain a distributed database. The BA pool provides strong guarantees of consistency via its distributed consensus algorithm. Thus writes or other modifications to the ledger or database only occur if a sufficient majority of pool members are in agreement. Consequently as long as only a minority of pool members are faulty or malicious at any time, modifications to the ledger and database will be trustworthy. Moreover, the process of entering transactions in either the ledger or pool results in the generation of confirming digital signatures on the contents of those transactions. The transactions can be verified by reading the contents of the database and ledger. The database will track the engagement and on-boarding of participants with the platform via their DIDs (Decentralized Identifiers) and associated key/ pairs. This database will also provide support for the remote management of computing devices by providing a signed code repository against which the configuration of any node can be verified. The database will also store capability and performance attributes for all the participating hosts including attribute based group network location and distance zones that will be used to suggest suitable federation groupings to consumers of compute. The pool will also maintain an order-book that matches available computing resource supply to requested demand. Once an order is matched the pool puts into escrow enough funds from the

consumer to pay for the ordered compute services. The escrow is released to the compute producers once they provide a proof of completion. The distributed consensus pool manages the settlement and finalization of orders as they are completed. All entities be they participants, hosts, orders, or any data item (DADi) are tracked with DIDs or derived DIDs (dDIDs) [34; 117]. This provides a universal mechanism for establishing the provenance of transactions flows in the system. The purpose is to provide credible processing chains in a decentralized control environment. This extends conventional centralized distributed streaming data architectures such as Kafka, Lambda, and Reactive programming to decentralized compute infrastructure in a secure and manageable way [55; 70; 79]. The distribution of credentials for marshaling serverless frameworks will also be based on the extensible properties of DIDs. Using the emerging W3C decentralized identifier (DID) standard as the basis for the distribution of credentials for marshaling ephemeral containers and the provenancing of data flows is a unique feature of Difuon. This infrastructure will support simplified client key management techniques for key reproduction, pre-rotation, and recovery. This identity layer enables credible reputations of participants on the platform.

The BA pool also preserves privacy by using a combination of group capability attributes and blinded routes. Group capability means that all members of the group share the same minimum capabilities. Thus the BA pool can make matches from the order book with a group of viable compute nodes that have the minimum capability for the transaction without indicating a specific member of the group. The requesting node then selects off-chain the specific member(s) from the group of potential providers it wishes to engage with for its compute tasks. An attacker would have to simultaneously exploit all the members of the group thereby making the exploit exponentially more expensive. A blinded route uses blinded signatures to provide verifiable but not publicly traceable network addresses to the potential participants in a future computing transaction [11; 30; 39; 134]. This is a type of zero-knowledge proof [150]. This is similar to how blinded signatures are used in E-Cash and E-Voting applications [16]. Thus each participant, source or sink of data, can verify that the route to each other provided by the BA pool is valid from the order book but the BA pool has no exploitable knowledge of the routes. As a result, the data may be transferred and the computation performed off-chain in a pair-wise private manner. Once the computation is complete, the exploitable value of the data exchanged is minimized. In addition, using self contained DADi based blockchains enables full end-to-end traceability and verifiability of a computation flow. Each member of a BA pool must be able to communicate with every other member. This is shown symbolically in the following diagram.

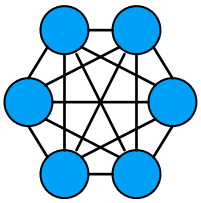

Fig. 2. Single BA distributed consensus pool.

To improve performance and minimize latency, multiple BA pools will be employed, each in a given zone grouped by network distance. Network distance may be measured by the round trip time, i.e. latency for a packet to traverse the network from one host to another and back again. Usually the network distance is a function of the number of router hops it encounters along the path. Because routers tend to be geographically distributed, using network distance also tends to result in a geographic distribution of BA pools. This is not exact as local data-centers may have high speed low latency hops to more remote data-centers. Often the latency is a function of the cost of the connection. Using network distance allows sources and sinks of data to be matched to minimize average latency and or cost or some combination of the two. The number of BA pools will expand as the network grows. The set of BA pools forms a distributed partially overlapping partial mesh of pools where the partial overlap between pools is with respect to that pool's nearest neighbors in network distance. The partial overlap provides redundant access by a given participant to multiple BA pools to increase availability and minimize common-mode failures. The partial mesh means that some pools can anchor their transactions with respect to some other pools but not all other pools. This is the chain-mail approach to increasing security without detracting from throughput [108]. In addition, the partial mesh means that some pools may make transactions that intersect participants between two different pools. This is a type of sharding. A hierarchy of intersecting pools will allow any set of participants to mutually engage in transactions albeit at increased latency. Fortunately the actual data transfer for compute happens pair-wise peer-to-peer off-chain and therefore its latency is not affected by the hierarchy of intersecting pools. The demand for cross pool transactions is minimized because the vast majority of fog computing interactions will benefit from nearer network locality. The following diagram shows a set of neighboring BA pools in a partial mesh where each interacts directly with its three nearest neighbors and can also interact with the intersection of a set of pools to enable indirect mutual interaction of all participants.

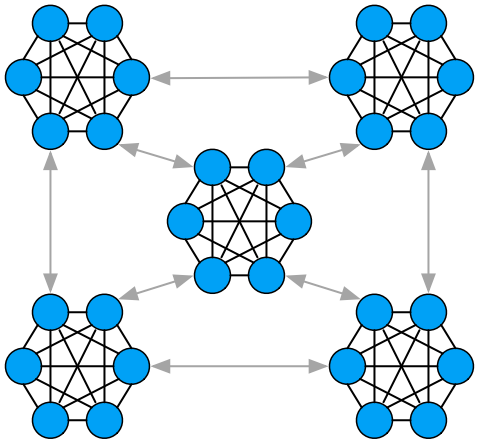

Fig. 3. System of geographically distributed BA pools

2.7.2. Relay Network

Augmenting the network of BA pools is a relay network of server nodes. One function of the relay network is provide read-only access to copies of the distributed ledgers and databases maintained by the BA pools. The relay network thereby reduces the read load on the BA pool members. As previously mentioned the main purpose of a BA pool is to provide a trustworthy verifiable mechanism for maintaining consistent updates to the distributed ledger and database managed by that pool. These data structures are designed such that a reader of a copy of the associated ledger or database can verify the consistency of the contents via the included signatures. Because the majority of accesses by users to the ledger and database will be read not write, the relay network improves scalability by both balancing the read traffic across the relay hosts and reducing the read traffic on the BA pool members. Because a relay node's copy of the ledgers and databases is read-only the only exploit a malicious relay node can perform is to delete or ignore updates to the associated contents but not modify those contents. Thus any user merely needs to be able to access one non-faulty relay node to be able to read and verify updates to the ledger and databases. This non-faulty relay node could be under the control of the user or some other trusted entity. Each user may select a set of relay nodes to provide higher availability. To restate, write updates require a trustworthy non-faulty majority. Whereas read verifies merely require a trustworthy minority – as little as one.

In addition to providing load balancing read/verify copies of the distributed ledgers and databases, relay nodes may support other functions. One of these is to provide a software defined data flow routing overlay [109]. This is similar to software defined networking but decentralized [119]. This maps data flow addresses to IP addresses. Thus a set of compute services or functions can be pre-configured with data flow addresses that only need to be converted to the real IP addresses of the actual hosts at run time. The relay nodes maintain a data flow route to IP routing table. This allows the serverless framework to dynamically allocate compute resources without reprogramming or reconfiguration of compute services and functions.

Another function of the relay network is to bootstrap peer-to-peer communications between nodes in the network by maintaining a DID-to-IP host:port mapping table. Most IP nodes in residential and commercial building networks do not have public IP addresses but have only private IP addresses and reside behind NAT routers. The default configuration of most NAT routers enables dynamic port mapping (forwarding) [96]. Dynamic port mapping opens a return port on the external (pubic) side of the NAT router as a result of an outgoing packet from the internal (private) side. This allows an internal host that does not have a public IP address to initiate two-way communications with an external host that does have a public IP address. It does not allow the converse. This is a problem for private hosts that want to work as compute servers. Private hosts can query the DID-to-IP address table on relay network hosts that have public IP addresses to helps bootstrap peer-to-peer communication with another private host. The basic technique is called UDP hole punching [137]. RAET is well suited to this approach. Hole punching is less reliable with TCP connections [130]. In the event that hole punching does not work, two nodes may communicate using the relay node as a repeater.

Another function of the relay network is to provide a content and code delivery network (CDN). Client side web applications require a static load of content. Having this static content cached in nodes that are nearby the web clients reduces startup delay and minimizes bandwidth demands. Moreover pushing updates to compute nodes code and frameworks can induce spikes in network traffic. This is minimize with cached local copies. Video and audio streaming applications consume large amounts of bandwidth this is also minimized if the content is cached locally.

Another function of a relay node is to facilitate federation of compute services. The relay nodes keep a local copy of the federation groups for compute services. The relay nodes track also track presence, availability, and performance of any of their connected or communicating nodes. This allows compute consumers to better dynamically select the specific servers to use from their federation pool.

In general the relay network's purpose is to facilitate and improve performance, horizontal scaling and greater elasticity of the computing infrastructure. The number of relay nodes is not fixed and may exceed the number of BA pool nodes by an order of magnitude. The following notional diagram shows a BA pool with auxiliary relay network nodes.

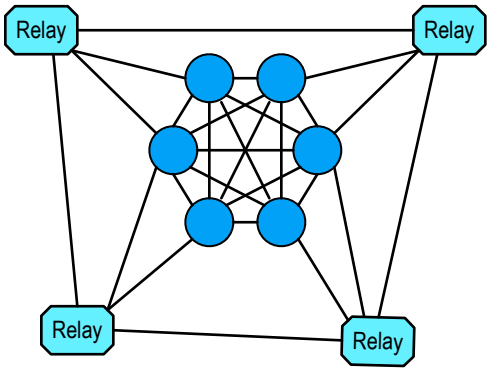

Fig. 4. Platform core made up of Relay network supporting a BA pool.

2.7.3. Compute, Storage, and App Network

The core the the Difuon platform is the distributed consensus control system with its supporting relay network. The rest of the platform is made up of participants in the platform's two sided marketplace. Platform participants have different roles. A participant role may be a consumer (user) or producer (provider) of services on the platform or both. They may pay for or get paid for services. A provider may also be the owner or controlling entity of other platform participants that are nodes hosting services. A user or provides pre-registers with its DID and any DIDs of any of its controlled participant nodes.

The main provider role is a *Fogger*. A *Fogger* is a self-provisioning, remotely managed, disposable compute server hosted in a residential and commercial building. *Foggers* provide public compute services via a serverless framework by either running micro-services or Functions-as-a-Service (FaaS). The provisioning and ongoing management of *Foggers* is executed by the platform core. Provisioning of a *Fogger* starts with preinstallation of the OS and associating it with its unique identifier (DID). Then once the *Fogger* is on site and powered up it attaches to the platform core and downloads its final configuration. It is then ready to accept compute jobs.

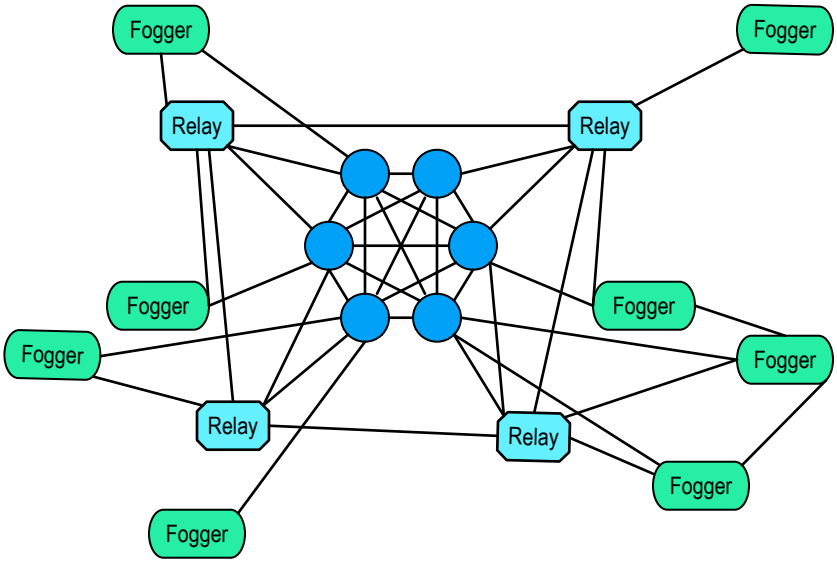

Fig. 5. Fogger network attached to Relay network and BA pool.

Another provider role is *Farmer*. A *Farmer* is a self-provisioning, remotely managed, disposable storage server hosted in a residential and commercial building. *Farmers* provide public persistent storage services. Although *Foggers* also provide storage, their storage is more ephemeral as is in direct support of a given serverless function or micro-service. In contrast, *Farmers*, provide persistent and/or archival storage. A single server can be both a *Fogger* and a *Farmer*. Like *Foggers*, the provisioning and ongoing management of *Farmers* is executed by the platform core. Provisioning of a *Farmer* starts with preinstallation of the OS and associating it with its unique identifier (DID). Then once the *Farmer* is on site and powered up it attaches to the platform core and downloads its final configuration. It is then ready to accept storage jobs.

Another provider role is *Podder*. A *Podder* is a self-provisioning, remotely managed, disposable application and storage server hosted in a residential and commercial building. *Podders* provide public Personal Online Datastores (PODs) i.e. personal fog [66; 75; 76; 86; 93; 120]. Unlike *Foggers* and *Farmers*, a *Podder* is an integrated set of services that support personal fog applications. Like other provider roles, the provisioning and ongoing management of *Podders* is executed by the platform core. Provisioning of a *Podders* starts with preinstallation of the OS and associating it with its unique identifier (DID). Then once the *Podder* is on site and powered up it attaches to the platform core and downloads its final configuration. It is then ready to accept provide personal fog application services. The following diagram shows the network with the addition of Farmers and Podders.

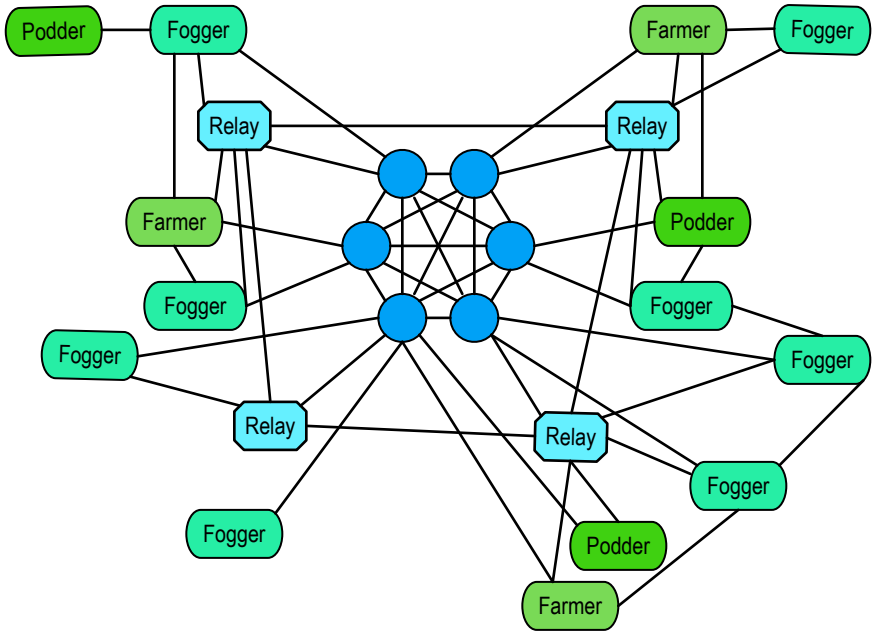

Fig. 6. Farmer, and Podder networks augmenting Fogger, Relay, and BA networks.

The *User* or *Consumer* roles are run by *Apps* (applications) hosted on other devices such as personal computers, smart phones, or cloud computers. These applications register with the Difuon network and have a DID. These applications interact with and make use of the services and functions running on the Foggers, Farmers, and Podders in the network. These might be applications that are used to support applications running on the Foggers. The following diagram shows the network with the addition of Users and Apps.

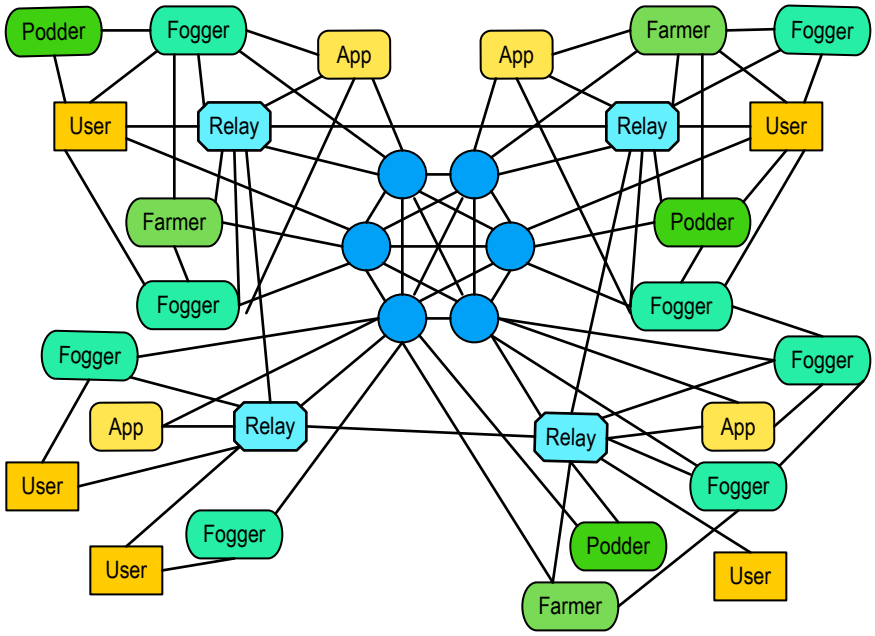

Fig. 7. Client applications and Users attached to infrastructure.

The final participant role are IoT devices such as sensors and actuators. These are sources or sinks of data that are input/output to the data integration and analysis apps running on the Foggers/Farmers. Because the Difuon Fog is an IP centric network, communication with IoT devices will be over IP. Non-IP IoT devices will need to communicate through gateways (not shown).The Relay nodes can help establish and maintain connections to the IoT devices and gateways.A full network with the IoT roles is shown in the following diagram.

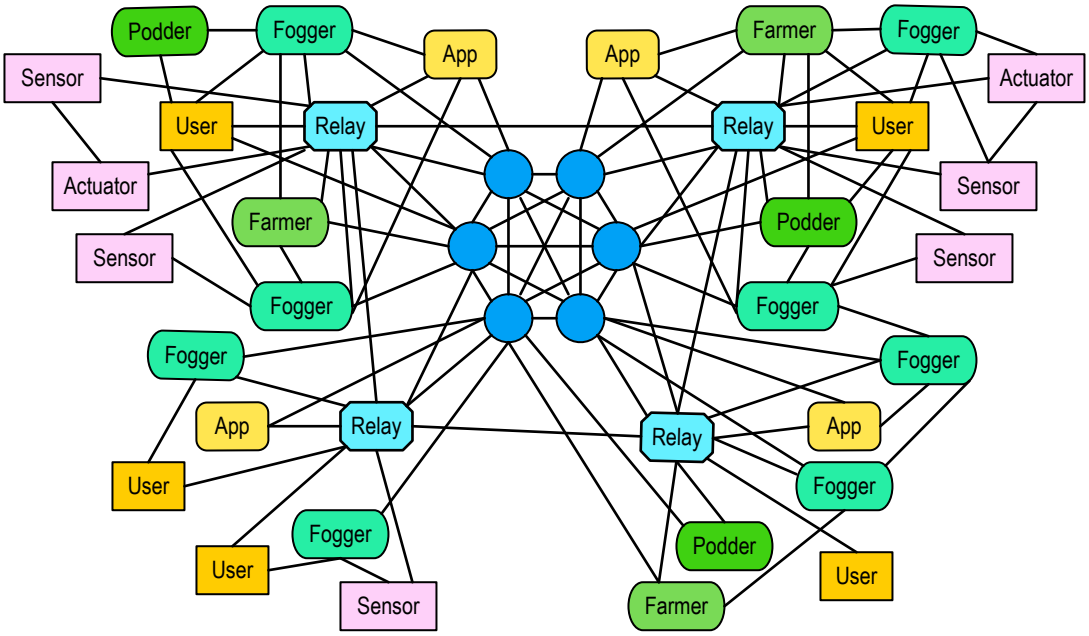

Fig. 8. Full system infrastructure with IoT sensors and actuators attached network with Users, Customer applications, Foggers, Farmers, Podders, Relayers, BA Pool members.

2.8. Business Model

The business opportunity is primarily leveraging blockchain technology, decentralized identity, cloud management software, and disposable servers to build a decentralized two-sided platform (network marketplace). This platform incentivizes producers to unleash their latent bandwidth, space, and energy with dedicated remotely managed servers at a near zero cost basis. This eliminates the main barrier to rapid build out of public fog computing infrastructure. Inherent in a decentralized approach is higher levels of privacy and security that coupled with convenient user experience and lower costs make public fog attractive to consumers. This combination is designed to drive two-sided network effects. The exponential two-sided network effects should eventually overcome scale and/or one-sided network effects head start that existing centralized cloud providers have. This business model combines the best of sharing economy and blockchain. Difuon's approach uses dedicated hardware that minimizes inconvenience and maximizes security and privacy. Our business model incentivizes users to host the hardware as fog compute nodes because users can monetize spare bandwidth, space, and energy. With our remote management technology these devices require no technical expertise by the user thereby enabling virtually anyone to participate. Our co-marketing arrangements with hardware manufacturers enable a scalable delivery model that will foster exponential growth of compute capacity in the fog with a near zero cost basis. In summary, decentralized public fog computing infrastructure monetizes spare space, energy, bandwidth with dedicated plug-n-play disposable servers running a secure, private, serverless (FaaS) platform that unleashes latent capacity in the edge with two-sided network effects at near zero cost basis!

2.8.1. Revenue Streams

The core revenue generators are as follows:

- Fees on marketplace transactional volume
- Fees for differentiated classes of service
- Fees on purchases from the application and software component store

The companion Difuon tokenomics white paper describes in much greater detail how dual the Store-of-Value (Sov) and Medium-of-Exchange (MoE) tokens work together to enhance positive feedback loops and positive cross-side network effects.

2.8.2. Use Cases

Although a public fog computing infrastructure can essentially supplant many of the compute and storage services that the public cloud provides there are some applications that are much better suited for public fog.

The simplest use case that is uniquely well suited for public fog is what we call the *personal fog*. The personal fog is like the personal cloud in terms of application but differs in that by virtue being hosted in the public fog is more secure, more private, and more self-sovereign [93]. Hosting personal cloud apps in a decentralized fog fosters self-sovereignty in that users have more control if not complete control over their personal data. This is much more compatible with the goals of the emerging *MyData* movement and the related Personal Online Datastore movement [66; 75; 76; 86; 120].

Another well suited use case are Web 3.0 decentralized applications (DApps) [5; 81]. Because the Difuon public fog is inherently decentralized it is easier to use it to host decentralized applications both web and others. Because it is based on a open portable decentralized identity standards (DIDs) it enables tighter integration with other applications based on the same standards.

On of the most uniquely well suited applications for a public fog computing infrastructure are IoT on-the-fly data aggregation and analytics applications. Because IoT data are sourced in the edge, a fog (edge) based computing infrastructure enables bottom

up processing that minimizes latency and transport costs while enhancing privacy, security, and flexibility. Indeed it is IoT data analytics and analysis that was the prime motivation for pursuing a decentralized fog computing infrastructure in the first place. The network effect benefits of IoT can best be realized when previously isolated IoT resources are enabled to interoperate. The well know primary barriers to unleashing the beneficial network effects of IoT are concerns about security, privacy, and agency. Due to a paucity of corresponding solutions, the IoT market delivers value well below its potential. A solution that only addresses one or two of these problematic barriers will be severely limited. Difuon provides a platform for federated ephemeral edge computing that viably solves all three problems of security, privacy, and agency.

Finally any private cloud application that requires levels of privacy and security that are not easily attained in a public cloud computing infrastructure might very well be better suited for the Difuon public fog computing infrastructure. In order to be decentralized and private and secure requires that the public fog use a diffuse trust perimeter-less security model. This model inherently provides mechanisms to enhance privacy, security, and control over applications and data whilst still being hosted on public compute infrastructure.

This approach is disruptive to the much more expensive cost basis of large centralized data-center hosted cloud services. Difuon takes a truly comprehensive best practices approach to the transforming compute infrastructure from centralized siloed one-sided networks to truly decentralized distributed two-sided networks. Only the latter can scale beneficially enabling the broadest adoption and participation of both producers and consumers of compute services.

3. Conclusion

The Difuon Fog exploits and integrates several emerging technologies to provide a decentralized computing infrastructure. for Internet of Things (IoT), Industrial Internet of Things (IIoT), Internet of Tiny Things (IoTT), and personal data applications at scale. The Difuon Fog provides a diffuse trust ephemeral computing trusted execution environment with end-to-end security (at motion and at rest) and modular privacy (off-chain interaction) that runs a distributed serverless framework (FaaS). This is essential enabling technology for a truly decentralized public fog computing infrastructure.

The distributed nature of IoT means that processing and analysis scales better if aggregated in a bottom up hierarchy from local to global. The data integration layer needs to be universal, not fragmented as it is now. It needs to leverage as much as possible the existing cloud based tooling for distributed applications but adapted to the unique characteristics of the fog.

Given the rampant fragmentation, stratification and siloization of existing IoT device level protocols, frameworks, and platforms, the only viable solution is an Internet Protocol (IP) centric approach that enables reliable, secure, decentralized integration across and through IoT silos and strata. This solution is a decentralized federated ephemeral edge computing framework, platform, and marketplace for data integration and analysis. This is the Difuon Fog.

Samuel M. Smith Ph.D. Is the CTO and co-founder of Difuon, a startup that is building a decentralized fog computing platform. He is advisory architect for reputation and AI for ConsenSys. He serves on the technical governance board of Sovrin.org. Samuel received a Ph.D. in Electrical and Computer Engineering from Brigham Young University in 1991. He then spent 10 years at Florida Atlantic University, eventually reaching full professor status. His research specialties include automated reasoning, machine learning, and autonomous vehicle systems. He has over 100 refereed publications in these areas and was principal investigator on numerous federally funded research projects. Dr. Smith has been an active participant in open standards development for networking protocols and is a serial entrepreneur.

References

[1] "About SONM,"
https://docs.sonm.com
[2] A. Back, M. Corallo, L. Dashjr *et al.*, "Enabling Blockchain Innovations with Pegged Sidechains," *Blockstream*,
https://blockstream.com/sidechains.pdf
[3] A. Opher, A. Chou, A. Onda *et al.*, "The Rise of the Data Economy: Driving Value through Internet of Things Data Monetization," *IBM Tech Report*, 2016
https://public.dhe.ibm.com/common/ssi/ecm/ww/en/www12367usen/global-business-services-cognitive-solutions-ww-white-paper-external-www12367usen-20180212.pdf
[4] A. Boyd, "The cloud computing market is about to get a lot more competitive," *InfoWorld*, 2018/02/14
https://www.infoworld.com/article/3255425/cloud-computing/the-cloud-computing-market-is-about-to-get-a-lot-more-competitive.html
[5] A. Hertig, "What is a Decentralized Application?," *Coiindesk*,
https://www.coindesk.com/information/what-is-a-decentralized-application-dapp
[6] A. W. Group, "OpenFog Architecture Overview," vol. OPFWP001.0216, 2016
https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf
[7] P.-L. Aublin, S. B. Mokhtar and V. Quéma, "Rbft: Redundant byzantine fault tolerance," vol. Distributed Computing Systems, no. ICDCS, pp. 297-306, 2013
[8] A. Asayag, G. Cohen, I. Grayevsky *et al.*, "Helix: A Scalable and Fair Consensus Algorithm Resistant to Ordering Manipulation," *Orbs.com*,
https://orbs.com/wp-content/uploads/2018/09/Helix-V1.3.pdf
[9] "AWS Serverless Application Model," *Amazon Lambda*,
[10] "Azure IoT Edge," *Microsoft*,
https://azure.microsoft.com/en-us/services/iot-edge/
[11] S. Baral, "An Efficient Blind Digital Signature Protocol Based on Elliptic Curve," *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 2 Issue 9, pp. 62-75, 2014
[12] B. Marr, "What Is Fog Computing? And Why It Matters In Our Big Data And IoT World," *Forbes*, 2016/10/14

https://www.forbes.com/sites/bernardmarr/2016/10/14/what-is-fog-computing-and-why-it-matters-in-our-big-data-and-iot-world/#1b0955c664ef
[13] “Bitcoin,” *Bitcoin.org*,
https://bitcoin.org/en/developer-documentation
[14] B. Butler, “What is fog computing? Connecting the cloud to things,” *Network World*, 2018/07/17
https://www.networkworld.com/article/3243111/internet-of-things/what-is-fog-computing-connecting-the-cloud-to-things.html
[15] “Byzantine fault tolerance,” *Wikipedia*,
https://en.wikipedia.org/wiki/Byzantine_fault_tolerance
[16] J. Camenisch, “Cryptographic e-Cash,” 2016
https://bitcoinschool.gr/slides/session5.pdf
[17] C. Coles, “Cloud Market in 2018 and Predictions for 2021,” *SkyHighNetworks*, 2018
[18] M. Carkci, “Dataflow and Reactive Programming Systems,” *Create Space Independent Publishing Platform*, 2014
[19] C. Schroder, “Marrying Ephemeral Docker Containers to Persistent Data,” *Linux.com*, 2016/09/15
https://www.linux.com/news/marrying-ephemeral-docker-containers-persistent-data
[20] M. Castro and B. Liskov, “Practical Byzantine fault tolerance,” vol. OSDI 99, pp. 173-186, 1999
[21] C. Mills, “Average internet speeds rose 23 percent last year,” *BGR*, 2018/07/10
https://bgr.com/2018/07/10/average-internet-speeds-us-vs-the-world/
[22] A. Clement, E. L. Wong, L. Alvisi *et al.*, “Making Byzantine Fault Tolerant Systems Tolerate Byzantine Faults.,” vol. NSDI 9, pp. 153-168, 2009
[23] “Cloud IoT Edge,” *Google*,
https://cloud.google.com/iot-edge/
[24] C. S. Alliance, “Custom Applications and IaaS Trends,” *McAfee Report*, 2018/04/01
[25] “CoAP: RFC 7252 Constrained Application Protocol,” *Coap.technology*,
http://coap.technology
[26] C. LeMahieu, “RaiBlocks: A Feeless Distributed Cryptocurrency
Network,” *Node RaiBlocks*,
https://raiblocks.net/media/RaiBlocks_Whitepaper__English.pdf
[27] C. Cotta, A. J. Fernández-Leiva, F. F. de Vega *et al.*, “Ephemeral computing and bioinspired optimization: Challenges and opportunities,” vol. Computational Intelligence, no. IJCCI, pp. 319-324, 2015
[28] “Crowd Computer,”
https://www.crowdmachine.com/crowdcomputer/
[29] D. Amodei and D. Hernandex, “AI and Compute,” *OpenAI Blog*, 2018/05/18
https://blog.openai.com/ai-and-compute/
[30] R. K. Das, S. K. Nayak, S. K. Bhoi *et al.*, “BSEA: A Blind Sealed-Bid E-Auction Scheme for E-Commerce Applications,” *Computers*, vol. 5, no. 4, pp. 32, 2016
[31] D. B. Stewart, “Software Components for Real Time,” *Embedded Systems Programming*, 2000/12/01
[32] D. Mazieres, “The Stellar Consensus Protocol:
A Federated Model for Internet-level Consensus,” 2016/02/26
https://www.stellar.org/papers/stellar-consensus-protocol.pdf
[33] D. Schwartz, N. Youngs and A. Britto, “The Rippple Consensus Algorithm,”
https://ripple.com/files/ripple_consensus_whitepaper.pdf
[34] “Decentralized Identifiers (DIDs),” *W3C Draft Community Group Report 23 August 2018*,
https://w3c-ccg.github.io/did-spec/
[35] “Delegated Proof-of-Stake Consensus,” *BitShares*,
https://bitshares.org/technology/delegated-proof-of-stake-consensus/
[36] J. Diaz-Montes, Y. Xie, I. Rodero *et al.*, “Federated Computing for the Masses--Aggregating Resources to Tackle Large-Scale Engineering Problems,” *Computing in Science & Engineering*, vol. 16, no. 4, pp. 62-72, 2014
[37] D. Stamat, “The Ephemeral Life of Dockerized Microservices,” *Iron*, 2015/01/22
https://blog.iron.io/the-ephemeral-life-of-dockerized/
[38] “Edge IoT Analytics,” *IBM*,
https://console.bluemix.net/docs/services/IoT/edge_analytics.html#edge_analytics
[39] E. H. El Kinani and F. Amounas, “Proposed Developments of Blind Signature Scheme based on The Elliptic Curve Discrete Logarithm Problem,” *Computer Engineering and Applications Journal*, vol. 2, no. 1, pp. 151-160, 2013
[40] E. Buchman, J. Kwon and Z. Milosevic, “The latest gossip on BFT consensus,” 2018/07/13
https://tendermint.com/docs/tendermint.pdf
[41] “Ethereum,” *Ethereum Foundation*,
http://www.ethdocs.org/en/latest/introduction/index.html
[42] FCC, “Sixth International Broadband Data Report,” 2018/02/02
https://transition.fcc.gov/Daily_Releases/Daily_Business/2018/db0209/DA-18-99A1.pdf
[43] P. Feldman and S. Micali, “An optimal probabilistic protocol for synchronous Byzantine agreement,” *SIAM Journal on Computing*, vol. 26, no. 4, pp. 873-933, 1997
[44] “Flow-based programming,” *Wikipedia*,
https://en.wikipedia.org/wiki/Flow-based_programming
[45] “Fog Computing,” *OpenFog Consortium*,
https://www.openfogconsortium.org/what-we-do/#definition-of-fog-computing
[46] “Fog computing for industrial automation,” *Consulting Specifying Engineer*, 2018/03/08
https://www.csemag.com/single-article/fog-computing-for-industrial-automation.html
[47] F. Research, “Developing a Framework to Improve Critical Infrastructure Cybersecurity,” *NIST*, 2013/04/08
https://www.nist.gov/sites/default/files/documents/2017/06/05/040813_forrester_research.pdf
[48] G. G. Edelman, “AI-powered ambient computing is just getting started,” *VentureBeat*, 2018/03/30
https://venturebeat.com/2018/03/30/ai-powered-ambient-computing-is-just-getting-started/
[49] “Getting started with Edge Analytics in Watson IoT Platform,” *IBM*, 2017/05/18
https://developer.ibm.com/recipes/tutorials/getting-started-with-edge-analytics-in-watson-iot-platform/
?cm_mc_uid=96956992698315366975959&cm_mc_sid_50200000=43260401537208012579
[50] Y. Gilad, R. Hemo, S. Micali *et al.*, “Algorand: Scaling byzantine agreements for cryptocurrencies,” vol. Proceedings of the 26th Symposium on Operating Systems Principles, pp. 51-68, 2017
[51] E. Gilman and D. Barth, “Zero Trust Networks: Building Secure Systems in Untrusted Networks,” “ O’Reilly Media, Inc.”, 2017.
[52] “Greengrass,” *Amazon AWS*,
https://aws.amazon.com/greengrass/

[53] I. Grigg, “The Ricardian Contract,” *Systemics*,
http://iang.org/papers/ricardian_contract.html
[54] I. Grigg, “Triple Entry Accounting,” *Systemics*, 2005
http://iang.org/papers/triple_entry.html
[55] I. Samizadeh, “A brief introduction to two data processing architectures—Lambda and Kappa for Big Data,” *Towards Data Science*, 2018/03/18
https://towardsdatascience.com/a-brief-introduction-to-two-data-processing-architectures-lambda-and-kappa-for-big-data-4f35c28005bb
[56] I. Lunden, “IBM to buy Red Hat for $34B in cash and debt, taking a bigger leap into hybrid cloud,” *TechCrunch*, 2018/10/28
https://techcrunch.com/2018/10/28/ibm-to-buy-red-hat-for-34b-in-cash-and-debt-taking-a-bigger-leap-into-hybrid-cloud/?utm_medium=TCnewsletter
[57] J. P. Morrison, “Flow-based Programming,” *jpaulmorrison.com*,
http://www.jpaulmorrison.com/fbp/
[58] J. Sanders and C. Forest, “Hybrid cloud: What it is, why it matters,” *ZDNet*, 2014/07/01
https://www.zdnet.com/article/hybrid-cloud-what-it-is-why-it-matters/
[59] J. D. Rey, “Amazon’s massive AWS outage was caused by human error,” *recode*, 2017/03/02
https://www.recode.net/2017/3/2/14792636/amazon-aws-internet-outage-cause-human-error-incorrect-command
[60] J. Engebretson, “iGR: Average Monthly Broadband Usage is 190 Gigabytes Monthly Per Household,” *TeleCompetitor*, 2016/09/26
https://www.telecompetitor.com/igr-average-monthly-broadband-usage-is-190-gigabytes-monthly-per-household/
[61] J. White, “Private vs. Public Cloud: What’s the Difference?,” *Expedient*, 2018
https://www.expedient.com/blog/private-vs-public-cloud-whats-difference/
[62] “Join us in building an open source decentralized identity ecosystem for people, organizations, apps, and devices.,” *Decentralized Identity Foundation (DIF)*,
http://identity.foundation
[63] J. Bonér, D. Farley, R. Kuhn *et al.*, “The Reactive Manifesto,” *reactivemanifesto.org*, 2014/09/16
https://www.reactivemanifesto.org
[64] J. Boner and V. Klang, “Reactive programming vs. Reactive systems,” *O’Reilly*, 2016/12/02
https://www.oreilly.com/ideas/reactive-programming-vs-reactive-systems
[65] J. Tsidulko, “The 10 Biggest Cloud Outages Of 2018 (So Far),” *CRN*, 2018/08/01
https://www.crn.com/slide-shows/security/300107391/the-10-biggest-cloud-outages-of-2018-so-far.htm
[66] “Journal of MyData,” *MyDataJournal Medium*,
[67] J. Jarosciak, “The future of cloud computing relative to becoming a monopoly,” *Blog*, 2017/09/29
https://www.joe0.com/2017/09/29/the-future-of-cloud-computing-relative-to-becoming-a-monopoly/
[68] J. V. Wagenen, “Connection Speed of Every State: 2018,” *StateTech*, 2018/05/08
https://statetechmagazine.com/article/2018/05/the-average-internet-connection-speed-of-every-state-2018-perfcon
[69] “Kinetic Edge,” *Vapor*,
https://www.vapor.io
[70] K. Malwski, “Why Reactive,” *O’Reilly*, 2017
https://www.oreilly.com/programming/free/why-reactive.csp?intcmp=il-webops-free-product-na_new_site_reactive_programming_vs_reactive_systems_text_cta
[71] L. E. E. M. O. N. BAIRD, “THE SWIRLDS HASHGRAPH CONSENSUS ALGORITHM: FAIR, FAST, BYZANTINE FAULT TOLERANCE,” 2016/05/31
https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf
[72] L. Tung, “GitHub rivals gain from Microsoft acquisition but it’s no mass exodus, yet,” *ZdNet*, 2018/06/06
[73] L. Columbus, “Roundup Of Cloud Computing Forecasts, 2017,” *Forbes*, 2017/04/29
[74] L. Columbus, “Forrester’s 10 Cloud Computing Predictions For 2018,” *Forbes*, 2017/11/07
https://www.forbes.com/sites/louiscolumbus/2017/11/07/forresters-10-cloud-computing-predictions-for-2018/#155557d4ae18
[75] E. Mansour, A. V. Sambra, S. Hawke *et al.*, “A demonstration of the solid platform for social web applications,” vol. Proceedings of the 25th International Conference Companion on World Wide Web, pp. 223-226, 2016
[76] M. Kutskova, ““MyData: nuts and bolts”,” *Medium MyData Journal*,
https://medium.com/mydata/mydata-nuts-and-bolts-f702a71c4a37
[77] J.-P. Martin and L. Alvisi, “Fast byzantine consensus,” *IEEE Transactions on Dependable and Secure Computing*, vol. 3, pp. 202-215, 2006
[78] M. Shacklett, “Edge Computing: A cheat sheet,” *TechRepublic*, 2017/07/21
https://www.techrepublic.com/article/edge-computing-the-smart-persons-guide/
[79] N. Marz and J. Warren, “Big Data: Principles and best practices of scalable real-time data systems,” New York; Manning Publications Co., 2015.
[80] M. Asay, “The future of serverless cloud looks a lot like physical servers,” *TechRepublic*, 2017/08/29
https://www.techrepublic.com/article/the-future-of-serverless-cloud-looks-a-lot-like-physical-servers/
[81] M. G. Zago, “Why the Web 3.0 Matters and you should know about it,” *Medium*, 2018/01/30
https://medium.com/@matteozago/why-the-web-3-0-matters-and-you-should-know-about-it-a5851d63c949
[82] M. Kanellos, “152,000 Smart Devices Every Minute In 2025: IDC Outlines The Future of Smart Things,” *Forbes*, 2016/03/03
[83] “Microservices,” *Wikipedia*,
https://en.wikipedia.org/wiki/Microservices
[84] M. Hearn, “Corda: A distributed ledger,” *corda.net*, 2016/11/29
https://www.corda.net/content/corda-technical-whitepaper.pdf
[85] M. I. T. M. Lab, “The IoT Data Tsunami Has Arrived, Now What?,” *Connected Things*, 2018/04/05
http://connectedthings.mitforumcambridge.org/overview/agenda-connected-things-2018/iot-data-tsunami/
[86] “My Data,” *MyData2018.0rg*,
https://mydata2018.org
[87] “Odroid C2,” *HardKernel*,
https://www.hardkernel.com/main/products/prdt_info.php?g_code=G145457216438
[88] “Open Portable Trusted Execution Environment,” *OP-TEE.org*,
https://www.op-tee.org
[89] O. Consortium, “Fog Computing,”
https://www.openfogconsortium.org/resources/
[90] P. Brogan, “USTelecom Industry Metrics and Trends 2018,” *USTELECOM*, 2018/03/01
https://www.ustelecom.org/sites/default/files/images/USTelecom%20Industry%20Metrics%20and%20Trends%202018.pdf
[91] P. Brogan, “New Analysis: USTelecom Data Highlights U.S. Broadband Expansion, Competition and Usage,” *USTELECOM*, 2018/03/20
https://www.ustelecom.org/blog/new-analysis-ustelecom-data-highlights-us-broadband-expansion-competition-and-usage
[92] P. Morrison, “Flow-Based Programming,” *flowbasedprogramming.com*, 2012/04/18
https://flowbasedprogramming.wordpress.com/article/flow-based-programming/

[93] “Personal Cloud,” *Wikipedia*,
[94] P. Bublies, “Data is giving rise to a new economy: How is it shaping up?,” *The Economist*, 2017/05/06
https://www.economist.com/briefing/2017/05/06/data-is-giving-rise-to-a-new-economy
[95] “Plenum Byzantine Fault Tolerant Protocol,” *HyperLedger Indy*,
https://github.com/hyperledger/indy-plenum/wiki
[96] “Port forwading,” *Wikipedia*,
https://en.wikipedia.org/wiki/Port_forwarding
[97] “Proof-of-stake,” *Wikipedia*,
https://en.wikipedia.org/wiki/Proof-of-stake
[98] “Proof-of-work system,” *Wikipedia*,
https://en.wikipedia.org/wiki/Proof-of-work_system
[99] “RandomizedConsensus,” *Yale*,
http://www.cs.yale.edu/homes/aspnes/pinewiki/RandomizedConsensus.html
[100] “Raspberry Pi 3 Model B+,” *Raspberry Pi*,
https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/
[101] R. G. Brown, “The Corda Platform: An Introduction,” *Corda.net*, 2018/05/01
https://www.corda.net/content/corda-platform-whitepaper.pdf
[102] “Rock64,” *Pine64*,
https://www.pine64.org/?page_id=7147
[103] T. Rocket, “Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies,”,” 2018
[104] R. Miller, “Edge computing could push the cloud to the fringe,” *TechCrunch*, 2017/08/03
https://techcrunch.com/2017/08/03/edge-computing-could-push-the-cloud-to-the-fringe/
[105] S. M. Smith, “Ioflo,” *Ioflo.com*,
https://github.com/ioflo/ioflo
[106] S. M. Smith, “Ioflo Documentation,” *Ioflo.com*,
https://github.com/ioflo/ioflo_manuals
[107] S. M. Smith, “Reliable Asynchronous Event Transport (RAET) Protocol,” *RaetProtocol*,
https://github.com/RaetProtocol/raet
[108] S. M. Smith, “Open Reputation Framework,” vol. Version 1.2, 2015/05/13
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/open-reputation-low-level-whitepaper.pdf
[109] S. M. Smith, “Many Cubed: Solving the Many-to-Many-to-Many Problem for Scalable Reliable Secure Distributed Internet Applications,” 2017/01/05
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/ManyCubed.pdf
[110] S. M. Smith, “Decentralized Autonomic Data (DAD) and the three R’s of Key Management,” Spring 2018
https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2018/blob/master/final-documents/DecentralizedAutonomicData.pdf
[111] S. M. Smith and D. Khovratovich, “Identity System Essentials,” 2016/03/29
[112] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008/10/31
https://bitcoin.org/bitcoin.pdf
[113] S. Fulton, “It’s a race to the edge, and the end of cloud computing as we know it,” *ZDNet*, 2017/11/14
https://www.zdnet.com/article/its-a-race-to-the-edge-and-the-end-of-cloud-computing-as-we-know-it/
[114] S. Popov, “The Tangle,” *Iota.org*,
https://iotatoken.com/IOTA_Whitepaper.pdf
[115] “Serverless Computing,” *Wikipedia*,
https://en.wikipedia.org/wiki/Serverless_computing
[116] S. Gaudin, “Update: 11-hour AWS failure hits websites and apps,” *Computerworld*, 2017/02/28
https://www.computerworld.com/article/3175643/cloud-computing/aws-failure-hits-web-sites-and-apps.html
[117] S. Conway, A. H., M. Ma *et al.*, “A DID for Everything,” *RWOT Fall 2018*, 2018/09/26
https://github.com/WebOfTrustInfo/rwot7
[118] S. Carew and N. Randewich, “Amazon touches $1 trillion, on pace to overtake Apple,” *Reuters*, vol. Business News, 2018/09/04
https://www.reuters.com/article/us-usa-stocks-amazon-com-trillion/amazon-touches-1-trillion-on-pace-to-overtake-apple-idUSKCN1LK1ZJ
[119] “Software-defined networking,” *Wikipedia*,
https://en.wikipedia.org/wiki/Software-defined_networking
[120] “Solid Explained,” *solid.inrupt.com*,
https://solid.inrupt.com/how-it-works
[121] S. Charara, “A quick and dirty guide to ambient computing (and who is winning so far),” *The Ambient*, 2018/01/04
https://www.the-ambient.com/features/quick-and-dirty-guide-ambient-computing-210-1-210
[122] S. Foundation, “Sovrin: Identity For All,”
https://sovrin.org
[123] “Sovrin: A Protocol and Token for Self- Sovereign Identity and Decentralized Trust,” *Sovrin.org*, 2018/01/01
https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Protocol-and-Token-White-Paper.pdf
[124] “Speedtest Global Index,” *SpeedTest*, 2018/08/01
http://www.speedtest.net/global-index
[125] S. Clair, “Truledger in Plain English,”
http://truledger.com/doc/plain-english.html
[126] Statistica, “Market capitalization of the biggest internet companies worldwide as of May 2018 (in billion U.S. dollars),” *Statistica*,
https://www.statista.com/statistics/277483/market-value-of-the-largest-internet-companies-worldwide/
[127] M. Steenstrup, M. A. Arbib and E. G. Manes, “Port automata and the algebra of concurrent processes,” *Journal of Computer and System Sciences*, vol. 27, no. 1, pp. 29-50, 1983
[128] D. B. Stewart, “Designing software components for real-time applications,” vol. Proceedings of Embedded System Conference, 2000
[129] M. Swan, “Blockchain: Blueprint for a new economy,” “ O’Reilly Media, Inc.”, 2015.
[130] “TCP hole punching,” *Wikipedia*,
https://en.wikipedia.org/wiki/TCP_hole_punching
[131] “The Ethash Algorithm,” *Ethereum Foundation*,
http://www.ethdocs.org/en/latest/mining.html#the-algorithm
[132] “The Future of Compute: A Global Market,” *ActiveAether*,
https://activeaether.com
[133] T. Bittman, “The Edge Will Eat the Cloud,” *Gartner Maverick Research*, 2017/09/22
https://www.gartner.com/doc/reprints?id=1-4GH0FTL&ct=171004&st=sb

[134] A. A. Thu and K. T. Mya, “Implementation of an efficient blind signature scheme,” *International Journal of Innovation, Management and Technology*, vol. 5, no. 6, pp. 443, 2014
[135] T. Hanke, M. Movahedi and D. Williams, “DFINITY Technology Overview SeriesConsensus System,” https://dfinity.org/pdf-viewer/pdfs/viewer?file=./library/dfinity-consensus.pdf
[136] “Triple-Signed Receipts,” *OpenTransactions.org*, http://opentransactions.org/wiki/index.php?title=Triple-Signed_Receipts
[137] “UDP hole punching,” *Wikipedia*, https://en.wikipedia.org/wiki/UDP_hole_punching
[138] V. Turner, “The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things,” *IDC Report*, 2014/04/01
[139] V. Turner, “Billions Of Sensors Making Millions Of Outcomes,” 2016/08/04 http://ecs.arrow.com/shared-assets/downloads/pdf/download_idc-billions-of-sensors.pdf
[140] V. Turner and C. MacGillivray, “IDC FutureScape: Worldwide IoT 2018 Predictions,” *IDC Report*, 2017/11/01 https://www.idc.com/getfile.dyn?sectionId=LISTOFATTACHMENTS&containerId=US43193617&elementId=US43193617-DL-0001&attachmentId=47282916
[141] V. Buterin, “The Meaning of Decentralization,” *Medium*, 2017/02/17 https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274
[142] V. Buterin and V. Griffith, “Casper the Friendly Finality Gadget,” https://github.com/ethereum/research/blob/master/papers/casper-basics/casper_basics.pdf
[143] Washo, “Ricardian Contracts in OpenBazaar,” *GitHub*,
[144] “WHAT IS A ZERO TRUST ARCHITECTURE?,” *Cyberpedia*, https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-architecture
[145] “Whitepaper:Nxt,” https://nxtwiki.org/wiki/Whitepaper:Nxt
[146] Wikipedia, “Fog Computing,” *Wikipedia*, https://en.wikipedia.org/wiki/Fog_computing
[147] Wikipedia, “Matrix Digital Rain,” *Wikipedia*, https://en.wikipedia.org/wiki/Matrix_digital_rain
[148] “Worldwide broadband speed league 2018,” https://www.cable.co.uk/broadband/speed/worldwide-speed-league/#regions
[149] Z. Witherspoon, “A Hitchhiker’s Guide to Consensus Algorithms,” *HackerNoon*, 2018/02/13 https://hackernoon.com/a-hitchhikers-guide-to-consensus-algorithms-d81aae3eb0e3
[150] “Zero-knowledge proof,” *Wikipedia*, https://en.wikipedia.org/wiki/Zero-knowledge_proof