2019 IEEE International Conference on Cloud Engineering (IC2E)

Addressing the Fragmentation Problem in Distributed and Decentralized Edge Computing: A Vision

Ketan Bhardwaj, Ada Gavrilovska, Vlad Kolesnikov
Matt Saunders, Hobin Yoon, Mugdha Bondre, Meghana Babu, Jacob Walsh
College of Computing, Georgia Institute of Technology, Atlanta, GA

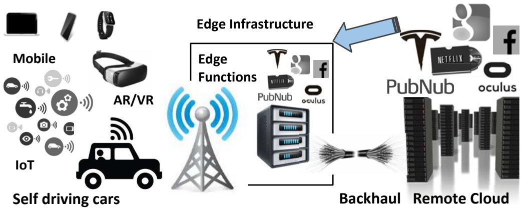

Figure 1: Vision of “Cloudified " Edge computing: Services supporting smart devices run across the edge and the cloud.

***Abstract*—At the core of the value proposition of edge computing is the ability to put computation close enough to the data sources, on demand. However, the data sources, computational infrastructure and software services needed to come together to power emerging and future edge computing applications are fragmented across different stakeholders, each with their own incentives, policies, and constraints on resources they can afford. This fragmentation limits the ability of edge computing to guarantee to applications and data *the edge* which will deliver the desired benefit. In this paper, we present our vision for an *Edge Exchange*, a decentralized directory service for a multi-stakeholder edge, as a path forward to enabling applications to be deployed across the best available edge resources, while still providing each stakeholder with controls regarding their resource use and sharing policies.**

I. The Fragmentation Problem in Edge Computing

Why Edge? Edge computing, i.e., the use of resources *distributed* at or near the devices, at the “cloudified” network edges as shown as shown in Figure 1, is becoming an additional key component of the end-to-end service infrastructure. Its fundamental benefits are tied to simple physics – speed of light and data movement energy. The ability to process data near its source, on the edges of the network, presents opportunities for reduced time-to-insight from that data.

The ability to reduce the distance traveled by data reduces the cost-of-insight. The potential for these benefits has generated a lot of excitement [1], [2], and has mobilized industry [3], [4], [5], [6], [7] and the research community [8], [9]. Many proof-of-concept demonstrations have been realized for different use cases, such as transportation [10], visual analytics [11], cognitive assistants [12], security [13], industrial controls [14]. Different software elements have been contributed across the stack, for the edge platforms [15], [16], [17], orchestration [18], [19], [20], middleware and data management [21], [22], etc.

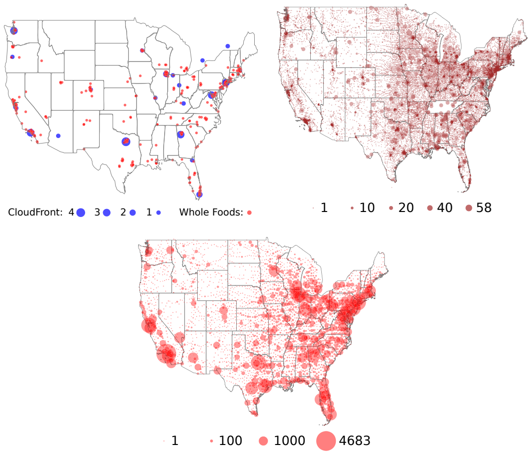

Figure 2: Potential Edge Locations

Where is the Edge? The currently viable edge infrastructure is what is deployed by enterprise cloud operators (e.g., Amazon’s CloudFront), or considered by mobile network operators at the cell phone towers or central stations. Figure 2.a[1] shows the AWS CloudFront edge servers (total 41) [23], plus potential future edge servers in WholeFoods locations (total 419) [26], (b) central office locations of mobile network operators (30,669), and (c) cell tower locations (217,346) [24], [25]. Different infrastructure heat-maps can be observed by analyzing the infrastructure footprint of other major players as well (Google [27] or Netflix [28]). The cell towers and central office locations shown belong to different mobile network operators (e.g., AT&T, Verizon, etc.). The cell towers are typically shared by multiple operators, and are owned by wireless infrastructure providers such as Crown Castle. Despite the presence of significant infrastructure investments made by the major players, there will be room for new providers. There are already new content cache providers delivering better cost and responsiveness for certain customers, compared to established players like Amazon and their CloudFront content caching service [29].

Can Edge Computing deliver? The above mentioned benefits that edge computing promises to afford cannot be realized unless the edge is pervasive. Having a pervasive edge is cost-prohibitive even for large corporations such as Microsoft, Amazon, Google, etc. As a result, not every

[1] Source [23] and FCC [24], [25] as of April, 2017.

978-1-7281-0218-4/19/$31.00 ©2019 IEEE
DOI 10.1109/IC2E.2019.00030

device is connected to the same edge provider, and not all edge providers are placed appropriately to deliver on the promised benefits. Furthermore, there are no inter-operable edge stacks available today that can work together seamlessly. Simply put, *the last-mile network over which the devices access today's services still remains fragmented* in terms of the mobile networks or ISPs which end users choose as their provider. This creates a number of hurdles for edge computing to succeed.

Contributions. In this paper, we make the case for new, presently missing technologies for cross-stakeholder resource orchestrations in edge computing. We present our vision for an *Edge Exchange*, a decentralized directory service, as a path forward to enabling applications to be deployed across the *best* available edge resources, even when those are fragmented across and owned by multiple parties, while still providing each stakeholder with controls regarding their resource use and sharing policies. We present the viability of this solution by reporting on our early experience with designing and prototyping select elements of an *Edge Exchange*.

II. Limitations

Fragmentation poses several constraints on the operation of the edge.

First, the data sources and end users benefiting from the edge applications are diverse due to their type, location and ownership. Second, the edge infrastructure also comes in diverse forms, with different capacity and performance characteristics. Third, edge computing applications (analytics, caching, ML inference, etc.) operate at different layers of software stacks. Fourth, all of these are disaggregated due to their ownership, physical location and networks that data sources or end users are connected to. Fifth, the resources are dynamic in nature due to multi-tenancy and varying network conditions. Finally, and perhaps most importantly, resource owners do not want to share the details of their edge offerings for security/privacy reasons, or to preserve their competitive advantages. Similarly, resource consumers also wish to retain control over the privacy of their data and applications.

As a result, edge computing is partitioned in application-specific, software-stack-specific, data source-specific and providers-specific silos. This falls short from a vision of edge computing where the edges of the network are "cloudfied", as shown in Figure 1. In summary, *fragmentation limits the ability of edge computing to guarantee to applications and data* the edge *which will deliver the desired benefit*; this in turn limits the benefits of edge computing and hinders its adoption. Such challenges in edge computing arise due to a number of inherent and interrelated factors, discussed below.

Choosing the right edge location *a priori*. The edge computing infrastructure is distributed and decentralized as evident in the location, placement and ownership of edge computing infrastructure discussed earlier. As a result, edge applications must strategically and explicitly select one or more among the available players, prior to deploying any applications. This does not scale as the number of providers starts to grow, especially considering mobility or resource churn [30].

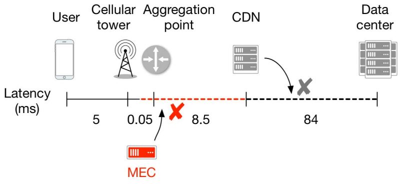

Figure 3: Potential latency afforded by different edge location.

Designing for edge performance *a priori*. The second challenge inherent in edge computing, is the diversity in performance that the edge can afford to applications. The edge is disaggregated at different points in the network, which impact the latency and available compute/storage resources. Figure 3 shows different end-user latencies from the edge, depending on where the user is located in the mobile network. As a result, edge applications must be intelligent and strategic about where the application is deployed, by choosing the right edge location with a correct latency profile and amount of resources needed by their applications. Doing this *a priori* strongly binds to a specific edge infrastructure, and makes it difficult to adapt to availability of different types of resources which may provide adequate performance.

Controlling edge, securing data and users *a priori*. The final challenge relates to the physical location of data sources with respect to end users. To provide guarantees in how data sources/end users are connected and can access edge computing applications, they must be part of a particular provider's network. For providers, it is counter-productive to invest in developing capabilities allowing users or data sources to migrate seamlessly across networks, as they wish to maintain a competitive advantage and to retain control of their edge. The real challenge is how to ensure that edge providers maintain *control and accountability* in their edge infrastructure without exposing their internal details. Similarly, applications seek controls for securing the data and/or their user information at the edge. Existing solutions do not suffice as they work in an all-or-nothing-sharing paradigm.

In summary, the key technical challenge is to remove *a priori* restrictions in the current status-quo, and to enable decentralized, just-in-time, secure, trusted bindings for application deployment in edge computing.

III. Vision for an Edge Exchange

The key objective toward broadening the impact and utility of edge computing, is to bridge across the boundaries of its many siloed stakeholders, so as to (1) provide control and accountability in how stakeholders interact and share resources, and to (2) enable applications to leverage the aggregate pools of edge resources and achieve the desired benefits.

We argue that this can be achieved by developing new decentralized systems, mechanisms and services which will enable cross-stakeholder coordinations to be performed dynamically, securely, and in a manner which enforces individual privacy and performance policies, as well as retains the benefits that edge computing is poised to deliver. What is needed are technologies that will bring down the existing barriers among disparate players at the edge, and facilitate controlled sharing of their data, infrastructure and services resource, thereby helping commoditize edge computing. For the edge computing stakeholders, this will provide controls and guarantees – contracts – over the access, sharing, or manipulation of their resources. For applications, this will provide flexibility and automation in determining the players to be involved in realizing the application's resource pool (i.e., services, infrastructure and data), in a way that honors individual policies and agreements, and continues to deliver the fundamental benefits of reduced latency and backhaul bandwidth usage. Such capabilities, particularly with adequate performance and efficiency, are missing from the current technology landscape, and the proposed research aims to address this gap.

Concretely, an *Edge Exchange* must address the following gaps in edge computing ecosystem.

- For the various stakeholders contributing resources, *Edge Exchange* must provide a new way for sharing resources across stakeholders without exposing internal details of those resources, and retaining the ability for each stakeholder to control and monitor their own resources' visibility, allocation, security and usage, ensuring stakeholder-specific policies are enforced.
- For the applications being deployed, *Edge Exchange* must provide new functionality which allows them to express, select, deploy and audit their performance and privacy goals.

These gaps can be addressed by designing an *Edge Exchange* as a new and efficient multi-stakeholder directory system, such as the one illustrated in Figure 4.

The *Edge Exchange* directory is a *decentralized* directory layer comprised of interconnected directory nodes. Each node represents a stakeholder, each with a different set of rules, policies and internal APIs.

The directory is *active* for two reasons. First, the interconnections among the directory elements are dynamically configured based on private attribute-based selection protocols. These determine the functionality which can be provided by *Edge Exchange*, i.e., the visibility or extent of edge resources that a given element in the directory affords. These protocols can be executed periodically, or in response to edge usage events (such as new applications). Second, the directory is active because in addition to simply serving authorized lookup information, its nodes dynamically resolve authorized and verified interfacing logic, which would permit an external party to bind to and interact with resources registered by another stakeholder at deployment time, as opposed to *a priori* deciding on it. This can be used to provide controlled access to data or to solve interoperability issues across stakeholders in a lazy fashion.

Edge Exchange-enabled interactions are *trusted*, providing multiple flavors of trust – from simply enabling the deployment of secure communication channels, to obfuscating the internal details of the interfaces, providing full vs. partial access to data, or both.

Finally, the interfaces and functionality provided by *Edge Exchange* enable new systems mechanisms which make it possible to *distribute applications across diverse and disaggregated edge resources*. Leveraging this new directory tier, *Edge Exchange* supports *look up* which operates across the *Edge Exchange* nodes and serves requests for resources needed to address specific performance or functional requirements of edge computing applications.

IV. An Edge Exchange Prototype Design

***Edge Exchange* abstractions.** To achieve the vision, *Edge Exchange* uses new abstractions to represent the capabilities and policies associated with different stakeholders, their various types of resources, their dependencies (data source connected to a gateway; edge server running several services, etc.), and their spatio-temporal relationships. The edge environment includes different types of resources that are distributed geographically and may have time-related constraints on their access. For instance, for *data sources* including sensors or actuators deployed as part of IoT, mobile and other personal devices, there is an inherent location component and they may be accessed at certain, but not all times. Similarly, *data processors* in form of hardware platforms, such as servers installed by AT&T and others, within their networks (at base stations, aggregation points, etc.), may be available at different locations and time. Finally, *data handlers* in form of higher-level services like Amazon's Greengrass or CloudFront, providing services such as IoT analytics, inferencing, or caching, may be deployed at strategic locations, but can be restricted in access at different times. This is illustrated in Figure 4 and tabulated in Table I. These resource elements correspond to *capabilities* in the edge environment.

A simple approach for characterizing independent stakeholders is to describe each of them as a separate namespace, within which they can use any arbitrary local naming convention to uniquely identify their capabilities. However, this works well only when there is a central authority that manages such namespaces, as employed in existing uses of active directories with a central authority [31]. Another approach involves use of directory services such as those used in peer-to-peer systems [32]. However, those systems typically use a flat namespace (based on content or routing).

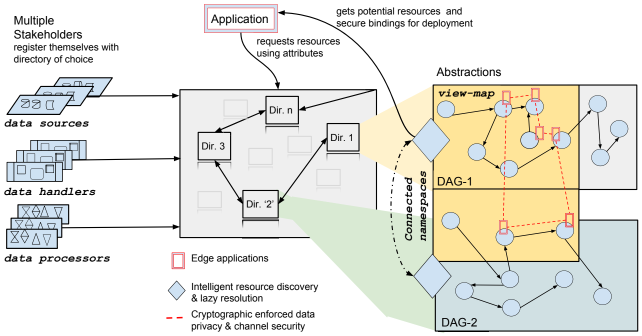

Figure 4: *Edge Exchange* overview showing edge application deployment across stakeholders.

abstraction	components of edge software stack
data source	"datasets" in form of files, time-series databases, content, parameters, etc.; "datastreams" in forms of pub-sub APIs, rest APIs that may include sensor values, cameras streaming video, etc.
data processors	"edge applications" implementing use case specific functions such as content caching algorithms, prerendering, inferencing, etc. in form of serverless functions, containers, etc.
data handlers	compute nodes in form of baremetal servers, container engines, serverless platforms

Table I: Different components in edge software stack abstracted by *Edge Exchange*.

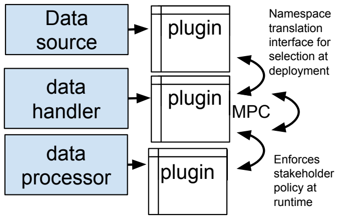

Figure 5: Interactions among *Edge Exchange* components.

As such, they cannot capture the relationships among different types of capabilities within a stakeholder. Using them can potentially lead to fragmentation of namespaces and siloed deployments, the original problem *Edge Exchange* aims to solve.

To address this, we propose a new concept of *connected typed namespaces*. Within a namespace, the different types of capabilities are arranged in DAGs which capture the relationships among them. A relationship may refer to an existing or possible connection between a data source and a data processor or handler. By incorporating this additional information in connected namespaces, we do not need more than one namespace for a single stakeholder to uniquely identify capabilities existing at different layers, e.g., hardware data processor vs. service providing data processing functionality. An *Edge Exchange* should support a set of basic operations on namespaces, such as querying, accessing, or mapping. Such interface points could be extended with stakeholder-specific policies which would enable *Edge Exchange* to integrate resource- and stakeholder-(namespace-)specific controls. The policies can be represented as a set of rules, a *contract* for the operations that will be integrated with resource accesses.

A single directory node in *Edge Exchange* can have multiple namespaces registered with it, representing different stakeholders. These, collectively, are part of the view of the edge visible to the applications or stakeholders at that location. To better serve latency-sensitive location-aware applications, each node may also represent this data as a *view map*, onto which namespaces are projected based on their geographic location. The combination of connected namespaces and view maps allows *Edge Exchange* to perform matching operation as per the requirements of applications without worrying about other entities in the system. For applications, this will create a simple interface to specify requirements and to access resources that meet those requirements. For stakeholders, this allows them to remain focused on their own capabilities and policies.

***Edge Exchange* operation.** From a bird's eye view, *Edge Exchange* is a set of interconnected directories hosted by any number of stakeholders in the edge environment. A stakeholder can choose to host their own directory or use another stakeholders' directory to offer their edge capability – their stake in the edge environments. The interconnections among those directories are either established explicitly (through external agreements among the stakeholders) or implicitly, through dynamic discovery and binding, continuously changing the view of the edge world visible to its stakeholders. Cross-directory interactions are driven by application requirements and are governed by stakeholder-specific policies.

Each *Edge Exchange* directory node contains information about the set of data sources, data handlers and data processors visible in a single *Edge Exchange* directory and offered by one or more stakeholders. The connections among the nodes are dynamically established. Stakeholders register with *Edge Exchange* through one or more *Edge Exchange* nodes, depending on trust, proximity, or other metrics. Each node is responsible for providing visibility and access control for the stakeholders and resources explicitly registered using it, based on stakeholder-specific policies. In this environment, the edge infrastructure can be seen as overlays combining different capabilities onto which application-specific components (or entire applications) are deployed. A single application may span across many of these capabilities, owing to their quality, availability or accessibility, as controlled by their stakeholders.

Applications request resources, in the form of data sources, data handlers and data processors through a trusted directory node in *Edge Exchange*, specifying requirements regarding performance, privacy, or data attributes. The discovery services determine exact or possible resource matches. The application may determine the final deployment configuration based on additional consideration of dependencies and placement constraints. The associated trust policies are evaluated and adequate secure bindings are established. The established mappings are (periodically) reevaluated and adjusted for changes in the environment, due to device churn, variability in load or connectivity, or due to explicit policy change.

In contrast to existing systems, *Edge Exchange* does not prescribes how each stakeholder offers their capabilities, nor the policies for control, privacy, and auditing, which may be needed for accountability. By doing so, *Edge Exchange* can integrate other complementary technologies being considered/developed for the edge computing space, for edge infrastructure and orchestration [18], [15], resource naming and discovery [33], privacy-centric data models [34], etc. An edge application or a part of it deployed on a single node has a data processor, data handler and/or data source, as shown in Figure 5. Stakeholders add *Edge Exchange* plugins to specify attributes, interfaces and policies associated with their access when adding resources to *Edge Exchange*. The individual elements are selected via private attribute-based matching and their respective policy (as defined by stakeholders) is enforced at deployment and runtime time via cryptographic primitives such as for multi-party computations (MPC) section VI supported via built-in *Edge Exchange* plugins.

***Edge Exchange* and MPC.** In the context of *Edge Exchange*, MPC as a standalone primitive will likely be available at the application as well as the service layer. In addition to providing privacy utility to applications and services, including MPC capabilities, we posit that it will allow us to better model, analyze and handle various system loads and demands. This is important because MPC is well on track to be ubiquitous, yet its network demands are relatively unexplored from the systems angle. Finally, we note that ultra-light MPC variants can be beneficially used even at the the lowest levels of system services.

The most important application of MPC in *Edge Exchange* will be its use in allowing clients to explore service offerings in a privacy-preserving manner. For example, a client who needs a certain minimal QoS (described as a set of parameter values) at a certain price, may determine, by running MPC with the provider, whether the latter is able to meet the needed requirements. This will all be done without revealing any information to either of the players, beyond the final output or a signed contract in case of the positive outcome.

While in general MPC may introduce significant overhead, existing MPC protocols can be tailored so as to achieve an acceptable cost/performance trade-off (or a set of trade-off points) [35], [36], [37]. Finding the right solution, however, requires exploration across the many MPC settings, such as semi-honest vs. malicious vs. covert, two-party vs. three-party vs. multiparty, or assumed maximum number of corruptions.

Summary of requirements. The requirements to be considered in building such a system are listed in Table II. While this list is not exhaustive it acts as a starting point to guide in developing the vision of *Edge Exchange*. These requirements fall into three main categories: functional requirements, system requirements and data requirements. Based on these requirements, we describe the early-stage implementation of select components on an *Edge Exchange*, described below.

V. Early Experience

Below, we summarize our observation that motivated this vision and our initial experiences with exploring the implementation of the requirements for an *Edge Exchange*.

A. *Motivating Experiment*

The goal of this experiment is to validate our hypothesis that *a priori* decision about choosing an edge node that satisfies given performance goals does suffice in edge computing. We show this by measuring the latency to different edge nodes from many providers. This gives us a better insight into the performance variability among individual providers at different geographic locations.

Experimental Setup. The experiment was conducted using the EdgeNet test network [38] which is a distributed edge cloud where nodes are provided by researchers across the world. These nodes are managed using Kubernetes which allows researchers to deploy containers easily to any edge location. We deployed a custom client to nine locations listed in the results below. We then measure the response latency to serverless functions we deployed in the three main cloud providers: Microsoft Azure, Amazon AWS, and Google Cloud. For each provider we made sure to deploy functions to different locations (including CDNs), and to use best available practices to reduce the RTT latency, such as by using the AWS's Lambda@Edge functionality.

The experiment ran for 36 hours during which each of our nine clients targets every serverless function across all locations and providers, one at a time. For each serverless instance we measure the RTT 30 times back-to-back, and repeated this process every 20 minutes. The measurements were collected using a centralized logging process, and the results are summarized in Figure 6. The results show that the latency from our edge clients varies across location, provider and time. For instance AWS may be better on average at one

Requirements	Description
Functional	• *Edge Exchange* must be open and freely accessible • Resource owners should be the only ones able to be add, modify, and remove their resources. • Resource owners are in charge of their own access policies • Resources should be reserved by only one user at a time. • Resource reservation must be easy to automate. • Add, remove, upgrade, downgrade user permissions • Add/remove resources to the network • Resource discovery and reservation
System	• Globally ordering transactions/updates to public information • Handling mobility and churn of nodes and minimizing the associated system latency • Stakeholder-specific policy creation and enforcement • Just-in-time secure, trusted bindings for applications and resources • Periodic reevaluation of application and device bindings
Data	• Each *Edge Exchange* node should store the following information • Local set of data sources, data handlers and data processors • Network Config • View-map • Stakeholder Config • *Edge Exchange* must manage the appropriate consistency and replication of the above data • Node storage footprint must be minimal

Table II: A summary of design requirements for realizing the *Edge Exchange* vision.

location while Azure may be better on average at another. Along with this, each providers RTT fluctuates over time meaning that while AWS may be best during some interval, Google Cloud may be better during the next one.

If services on the edge are going to connect in an on-demand way we must provide ways for services to search for the *best match*, and this *best match* must be periodically recomputed. This is much different than the cloud environment where these bindings are established *a priori* and remain enforced for the duration of the process. The results also serve as a validation in our hypothesis regarding the need for an *Edge Exchange*, and motivate the further design exploration to create decentralized stack for edge exchanges.

B. Implementation exploration

In order to gain insights into the tradeoffs in the design and implementation space concerning *Edge Exchange* and its abstractions, we chose to initially use several existing software technologies. Here we present two of these early efforts:

Dynamic namespace management with Apache Zookeeper. We designed and implemented a prototype of an active directory service using Apache Zookeeper [39]. Zookeeper (ZK) provides us with some of the functionality needed for *Edge Exchange*. For instance, we use ZNodes to represent elements in the distributed directory, and the ZK naming service for resource encoding. Specifically, ZK version 3.5.4-beta provides dynamic reconfiguration of the ensembles, which we can employ to represent the dynamic nature of *Edge Exchange*. Currently, we have prototyped a deployment of a dynamic configuration, or ensemble of ZK servers, which represent a *Edge Exchange* node, and a test application which connects to the *Edge Exchange* node (as a ZK client). For the applications trying to connect to a resource, we use a 2-phase protocol to handle concurrent requests. The protocol implements a preliminary find and match functionality to obtain a resource which matches the application's requirements, and can form the basis for the dynamic directory configuration and the resource allocation protocol.

Figure 7 illustrates a sample result from the evaluation of the ZK-based implementation of the namespace management functionality. The bars shows the duration of operations for registering resources with an *Edge Exchange* in a system with multiple concurrent participants issuing requests. Deeper investigation determines that the Join Ensemble operation in ZK dominates the latency of Device Join by consuming 86.6% of the total time. This behavior is expected in the case of concurrent device join requests as devices contend to change the ZK ensemble configuration to add themselves to the ensemble. In the case of concurrent requests, ZK allows one request to proceed, while others encounter an exception and back off with a randomized timeout between 0ms to 100ms. Because of the concurrent requests, we see a variation in Join Ensemble from 36ms to 2056ms for the fairly modest degree of concurrency experienced in this testbed. In the case of single requests, the Join Ensemble phase varies from 37ms to 96ms. The results illustrate that although adequate for providing the functionality for dynamic namespace changes, a system with strong consistency guarantees, such as Zookeeper, will not support the namespace management functions for an *Edge Exchange*. Instead, introducing abstractions such as views provide a

	Locations
Serverless	AWS Lambda@Edge, Azure - Central, Azure - East, Azure - East 2, Azure - North Central, Azure - South Central, Azure - West, Azure - West 2, Azure - West Central, GCloud - Central, GCloud - East
EdgeNet	carleton, edgenet1, edgenet2, gpo, nps, toronto, toronto-core, washington, waterloo

Table III: Locations of serverless infrastructure and EdgeNet servers used in experiments

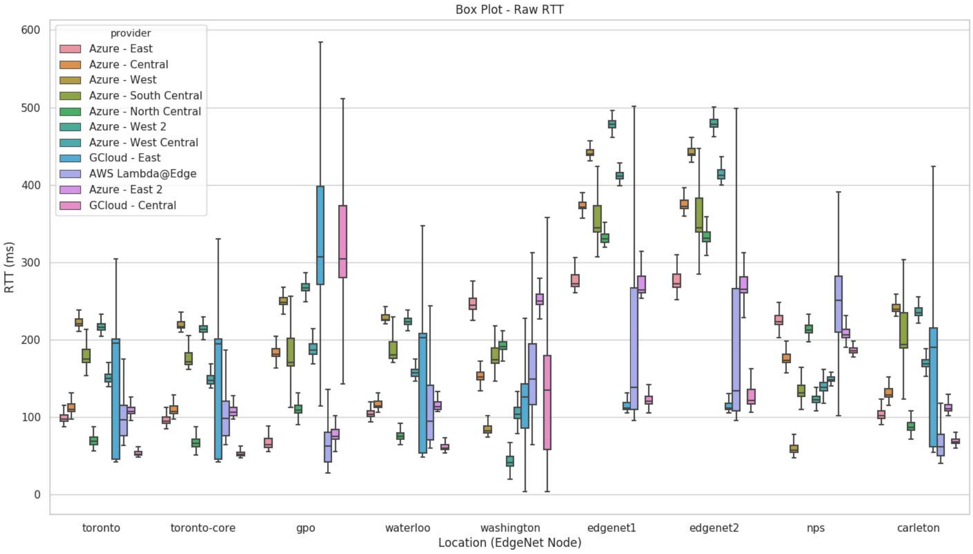

Figure 6: RTT between different serverless providers as observed from different EdgeNet locations

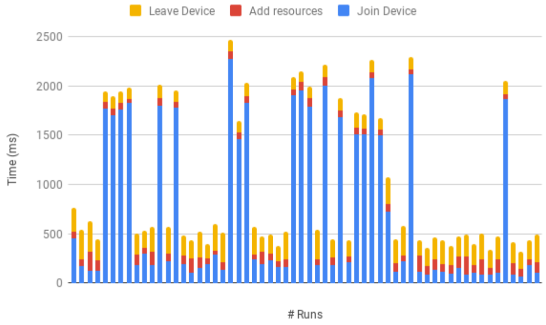

Figure 7: Latency of different phases of a device's lifetime

mechanism to bound the domains across which consistency is required, and deliver on both – functionality *and* performance.

Decentralized control with Hyperledger Fabric. The second approach we are exploring uses the blockchain-based Hyperledger Fabric (HLF) technology [40] to create a distributed directory. An *Edge Exchange* node is represented by a full HLF deployment which contains at least one organization. An organization translates to an individual stakeholder who would like to add resources to the shared *Edge Exchange* node ledger. Each organization is able to then use their own membership service provider in order to validate and control their own resources and manage their own ledger. Stakeholders are able to share and communicate information using the concept of channels which helps support the shared *Edge Exchange* node ledger. Policies can be supported via HLF's built-in support for smart contracts [40] which allow for immutable complex agreements to be upheld in a decentralized manner. Our current working system allows us to deploy an HLF system simulating an *Edge Exchange* node consisting of any number of stakeholders using containerization tools including Docker and Kubernetes. This provides maximum portability and efficiency when deploying *Edge Exchange* nodes.

HLF proves to be a promising approach as it addresses many scalability concerns related to blockchain technologies, by avoiding POW and by introducing the concept of channels which help partition the ledger data and reduce communication overhead. HLF is extremely modular, allowing for customization and growth as new technologies are developed. HLF supports both CouchDB and LevelDB allowing for rich queries on the ledger. Transactional throughput and performance under a large amount of node churn is still the main concern in this blockchain based approach.

Further Limitations. As stated above, we have prototyped partial functionality of *Edge Exchange* using data processors that are based on Kubernetes [41] to provide container-based capabilities. Although useful to speed up the initial development of *Edge Exchange*, Kubernetes is limited for our use case in multiple ways. For instance, it is designed to orchestrate only a single cluster with nodes located in a single data center; instead, *Edge Exchange* will need to operate across a geographically distributed and diverse set of edge resources. Also, Kubernetes is location-agnostic so it cannot readily support location-based policies to be used for tasks such as orchestration. Similar problems exist with serverless frameworks such as OpenLambda [42]. Policy support in serverless frameworks can possibly be prototyped using their permission model [43]. However, they are limited in that they cannot support fine-grained policy specification and/or enforcement. Despite their limitations, these open source technologies, and our preliminary implementation of the rudimentary prototypes, provide a starting point in our exploration of the design space for an *Edge Exchange*.

VI. Related work

There are a number of efforts which provide elements of the technologies required for the *Edge Exchange* vision. Seattle [44] is a publicly accessible edge computing platform to enable real-world application deployment on heterogeneous nodes. It addresses the lack of open interfaces in edge computing infrastructure by providing pre-configured containers for the edge. Seattle provides a trust management intermediary in the form of clearinghouse which is a human interface to request and deliver sandboxes to run edge applications.

Although there has been significant research in the general area of security and privacy in edge and fog computing [45], [46], [47], [48], [49], [50], literature on access management for resource sharing is rather scarce. In order to improve the efficiency of resource lookup using centralized architectures, distributed and peer-to-peer networks, RESTful web APIs, etc., include approaches that employ centralized search engines [51], single [52] or k-hop [53] distributed hash tables, and other scalable look up solutions [54], [55], [56]. Although relevant, these are not applicable to an edge computing infrastructure without significant modifications. More recently, Kinaara [57], proposes to add discovery capabilities in edge computing and uses discrete binary representations for resources combined with DHT-like [58] lookup, but requires a mediator.

Other related work towards resource management in edge computing involves techniques from game theory [59], [60], cooperative management [61], multiple tiers involving the benefits of a cloud-tier [62], and linear programming [63].

FocusStack [64] uses a primitive called GeoCast [65] to allow discovery of nearby IoT devices and edge-compute resources independently, using their geographical location and resource types. SWORD [66] provides a decentralized resource discovery service that allows clients to select nodes from PlanetLab using the Vivaldi network coordinates protocol [67]. NodeFinder [68] is a scalable mechanism for searching across geo-distributed state.

None of them, however, support departure from their own prescribed schemes on how and what can be offered, and are therefore unable to provide edge computing stakeholders with distinct and fine-grained control over their resources and policies. The vision we present in this paper is that there is a need for a future system such as *Edge Exchange* to address this gap.

Secure multi-party computation (MPC), also sometimes referred to as computing under encryption, is a very active research subarea of cryptography. MPC enables two or more distrustful parties to jointly evaluate any function on their private inputs without revealing anything except for the result itself. Seen in generality, MPC encompasses nearly any network functionality one can imagine – any kind of data transfer, computation, communication, – while guaranteeing the above data privacy property. Depending on the specifics of the computed function, the overhead of MPC over plaintext computing can vary from negligible to unacceptable. As MPC tools, both generic (e.g. Garbled Circuit (GC) [69], [70], [71], [72], [73], [74], [75], [76], [77], [35], [36], [78]) and tailored (e.g., Private Set Intersection (PSI) [79], [80], zero-knowledge proofs [37] and Private DB querying [81], [82]) have dramatically increased in performance in the recent years, many privacy enforcing computations became within reach. *Edge Exchange* can adapt these advancements and make them applicable in edge computing.

VII. Conclusions

We discuss an important problem for edge computing: infrastructure, services, and data in the last mile of the network are fragmented across mobile networks and ISPs, making it difficult to guarantee *the edge* required by applications. One way to solve this is through intermediaries, as what is recently done to enable multi-cloud application deployments [83], [84]. Multi-cloud solutions wrap interfaces offered by each cloud provider with a common set of APIs that can be used for deploying applications in cloud-agnostic manner. However, prescribing even a common API to every stakeholder is a problem that is impossible to solve with a technical solution alone. There are already signs that this is very hard to achieve even for a few cloud players. In edge computing, where the number of stakeholders are expected to be much larger, it will become a daunting task. In response, we present a vision of a decentralized, distributed, active, and trusted directory service, as a key enabler for cross-stakeholder resource management and orchestration. The goal is to make it possible to support rich types of cross-player interactions needed to allow and ensure resources at the edge are combined in a way that maintains performance and privacy requirements. We describe the design of a prototype *Edge Exchange* system, its key requirements and abstractions, and briefly discuss our initial experiences.

References

[1] P. Levine, “The End of Cloud Computing,” https://a16z.com/2016/12/16/the-end-of-cloud-computing/.

[2] "The Edge will eat the Cloud," https://blogs.gartner.com/thomas_bittman/2017/03/06/the-edge-will-eat-the-cloud/.

[3] "Openfog consortium," https://www.openfogconsortium.org/.

[4] "Etsi mobile edge computing," http://goo.gl/Qef61X.

[5] "Intel network edge virtualization," https://networkbuilders.intel.com/network-technologies/nev.

[6] "Vmware pulse iot center," https://www.vmware.com/products/pulse.html.

[7] P. Rodriguez, "The Edge: Evolution or Revolution," in *ACM/IEEE Symposium on Edge Computing (SEC'17)*, 2017.

[8] "ACM IEEE Symposium on Edge Computing," http://acm-ieee-sec.org/.

[9] "USENIX Workshop on Hot Topics in Edge Computing ," https://www.usenix.org/conference/hotedge18.

[10] K. Lee, J. Flinn, and B. D. Noble, "Gremlin: Scheduling interactions in vehicular computing," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, ser. SEC '17. New York, NY, USA: ACM, 2017, pp. 4:1–4:13. [Online]. Available: http://doi.acm.org/10.1145/3132211.3134450

[11] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasiman, "Cachier: Edge-Caching for Recognition Applications," in *IEEE Int'l Conf on Distributed Computing Systems (ICDCS)*, 2017.

[12] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 68–81. [Online]. Available: http://doi.acm.org/10.1145/2594368.2594383

[13] K. Bhardwaj, J. C. Miranda, and A. Gavrilovska, "Towards iot-ddos prevention using edge computing," in *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*. Boston, MA: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/hotedge18/presentation/bhardwaj

[14] C. H. Chen, M. Y. Lin, and C. C. Liu, "Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers," *IEEE Network*, vol. 32, no. 1, pp. 24–32, Jan 2018.

[15] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing '09*, 2009.

[16] K. Bhardwaj, M.-W. Shih, P. Agarwal, A. Gavrilovska, T. Kim, and K. Schwan, "Fast, scalable and secure onloading of edge functions using airbox," in *Proceedings of the 1st IEEE/ACM Symposium on Edge Computing (SEC'16)*, Washington, D.C., 2016.

[17] D. F. Willis, A. Dasgupta, and S. Banerjee, "Paradrop: A multi-tenant platform for dynamically installed third party services on home gateways," in *Proceedings of the 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing*, ser. DCC '14. New York, NY, USA: ACM, 2014, pp. 43–44. [Online]. Available: http://doi.acm.org/10.1145/2627566.2627583

[18] "OpenEdge Computing," http://openedgecomputing.org/.

[19] "Edgex roundary," https://www.edgexfoundry.org/.

[20] "Open reference for 5g innovation," https://opencord.org/m-cord-open-reference-solution-paves-the-way-for-5g-innovation/.

[21] H. Gupta, Z. Xu, and U. Ramachandran, "DataFog: Towards a Holistic Data Management Platform for the IoT Age at the Network Edge," in *Proceedings of the 1st USENIX Conference on Hot Topics in Edge Computing*. USENIX Association, 2018.

[22] U. Ramachandran, R. S. Nikhil, J. M. Rehg, Y. Angelov, A. Paul, S. Adhikari, K. M. Mackenzie, N. Harel, and K. Knobe, "Stampede: A cluster programming middleware for interactive stream-oriented applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 11, pp. 1140–1154, 2003.

[23] "AWS CloudFront Locations," https://aws.amazon.com/cloudfront/details.

[24] "Antenna structure registration," http://wireless.fcc.gov/antenna/index.htm?job=home.

[25] "Marigold direct: List of central offices," http://www.marigoldtech.com/lists/co.php.

[26] "Whole Foods Locations," https://www.wholefoodsmarket.com/stores/list/state.

[27] "Google Edge Network, Infrastructure for Google Global Cache (GGC)," https://peering.google.com/infrastructure.

[28] "Researchers Map Locations of 4,669 Servers in Netflix's Content Delivery Network," https://spectrum.ieee.org/tech-talk/telecom/internet/researchers-map-locations-of-4669-servers-in-netflixs-content-delivery-network.

[29] R. Biehl, "Amazon CloudFront – Why More Isn't Always Better," 2016, medium.com/@robertbiehl/amazon-cloudfront-why-more-isn-t-always-better-100bd8d5fa20.

[30] K. Habak, M. Ammar, K. Harras, and E. Zegura, "Femto-Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge," in *Proceedings of the 8th IEEE International Conference on Cloud Computing*, 2015.

[31] "Microsoft active directory system overview," https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/WinArchive/[MS-ADSO].pdf.

[32] A. Malatras, "State-of-the-art survey on p2p overlay networks in pervasive computing environments," *Journal of Network and Computer Applications*, vol. 55, pp. 1 – 23, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804515000879

[33] "Named data networking," https://named-data.net/.

[34] R. Spahn, J. Bell, M. Lee, S. Bhamidipati, R. Geambasu, and G. Kaiser, "Pebbles: Fine-Grained Data Management Abstractions for Modern Operating Systems," in *Proceedings of the USENIX Operating Systems Design and Implementation (OSDI)*, Broomfield, CO, 2014.

[35] X. Fan, C. Ganesh, and V. Kolesnikov, "Hashing garbled circuits for free," in *Advances in Cryptology – EUROCRYPT 2017, Part III*, ser. Lecture Notes in Computer Science, J. Coron and J. B. Nielsen, Eds., vol. 10212. Springer, Heidelberg, Apr. / May 2017, pp. 456–485.

[36] X. Wang, A. J. Malozemoff, and J. Katz, "Faster secure two-party computation in the single-execution setting," in *Advances in Cryptology – EUROCRYPT 2017, Part III*, ser. Lecture Notes in Computer Science, J. Coron and J. B. Nielsen, Eds., vol. 10212. Springer, Heidelberg, Apr. / May 2017, pp. 399–424.

[37] J. Katz, V. Kolesnikov, and X. Wang, "Improved non-interactive zero knowledge with applications to post-quantum signatures," in *ACM CCS 18: 25th Conference on Computer and Communications Security*, D. Lie, M. Mannan, M. Backes, and X. Wang, Eds. ACM Press, Oct. 2018, pp. 525–537.

[38] J. Cappos, M. Hemmings, R. McGeer, A. Rafetseder, and G. Ricart, "EdgeNet: A Global Cloud that Spreads by Local Action," in *ACM Symposium on Edge Computing (SEC'18)*, 2018.

[39] P. Hunt, M. Konar, F. P. Junqueira, and B. Reed, "Zookeeper: Wait-free coordination for internet-scale systems," in *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, ser. USENIXATC'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 11–11. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855840.1855851

[40] "Hyperledger," https://www.hyperledger.org/.

[41] D. K. Rensin, *Kubernetes - Scheduling the Future at Cloud Scale*. 1005 Gravenstein Highway North Sebastopol, CA 95472: O'Reilly and Associates, 2015. [Online]. Available: http://www.oreilly.com/webops-perf/free/kubernetes.csp

[42] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Serverless Computation with OpenLambda," in *8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)*. Denver, CO: USENIX Association, 2016. [Online]. Available: https://www.usenix.org/conference/hotcloud16/workshop-program/presentation/hendrickson

[43] "Permissions in amazon greengrass server less," https://docs.aws.amazon.com/lambda/latest/dg/intro-permission-model.html.

[44] A. Rafetseder, L. Pühringer, and J. Cappos, "Practical Fog Computing With Seattle," in *Fog World Congress*, 2017.

[45] C. Dsouza, G. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, Aug 2014, pp. 16–23.

[46] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurr. Comput. : Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, Jul. 2016. [Online]. Available: https://doi.org/10.1002/cpe.3485

[47] F. Li, Y. Rahulamathavan, M. Conti, and M. Rajarajan, "Robust access control framework for mobile cloud computing network," *Comput. Commun.*, vol. 68, no. C, pp. 61–72, Sep. 2015. [Online]. Available: https://doi.org/10.1016/j.comcom.2015.07.005

[48] B. Zaghdoudi, H. K. Ayed, and W. Harizi, "Generic access control system for ad hoc MCC and fog computing," in *CANS*, ser. Lecture Notes in Computer Science, vol. 10052, 2016, pp. 400–415.

[49] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Secure data sharing and searching at the edge of cloud-assisted internet of things," *IEEE Cloud Computing*, vol. 4, no. 1, pp. 34–42, Jan 2017.

[50] L. Popa, M. Yu, S. Y. Ko, S. Ratnasamy, and I. Stoica, "Cloudpolice: Taking access control out of the network," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: ACM, 2010, pp. 7:1–7:6. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868454

[51] S. K. Datta, R. P. F. D. Costa, and C. Bonnet, "Resource discovery in internet of things: Current trends and future standardization aspects," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 542–547.

[52] R. Rodrigues, B. Liskov, and L. Shrira, "The design of a robust peer-to-peer system," in *Proceedings of the 10th Workshop on ACM SIGOPS European Workshop*, ser. EW 10. New York, NY, USA: ACM, 2002, pp. 117–124. [Online]. Available: http://doi.acm.org/10.1145/1133373.1133396

[53] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse, "Kelips: Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead," in *IPTPS'03*, 2003.

[54] "Kazaa media desktop." http://www.kazaa.com/.

[55] "Gnutella." http://www.gnutella.com.

[56] A. T. Mýzrak, Y. Cheng, V. Kumar, and S. Savage, "Structured superpeers: Leveraging heterogeneity to provide constant-time lookup," in *Proceedings of the The Third IEEE Workshop on Internet Applications*, ser. WIAPP '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 104–. [Online]. Available: http://dl.acm.org/citation.cfm?id=832311.837395

[57] A. Salem, T. Salonidis, N. Desai, and T. Nadeem, "Kinaara: Distributed discovery and allocation of mobile edge resources," in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2017, pp. 153–161.

[58] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '01. New York, NY, USA: ACM, 2001, pp. 149–160. [Online]. Available: http://doi.acm.org/10.1145/383059.383071

[59] Y. Sun and N. Zhang, "A resource-sharing model based on a repeated game in fog computing," *Saudi Journal of Biological Sciences*, vol. 24, no. 3, pp. 687 – 694, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319562X17300529

[60] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, "A hierarchical game framework for resource management in fog computing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 52–57, Aug 2017.

[61] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, and Y. Zhang, "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, Dec 2015.

[62] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "Enorm: A framework for edge node resource management," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.

[63] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.

[64] B. Amento, B. Balasubramanian, R. J. Hall, K. Joshi, G. Jung, and K. H. Purdy, "Focusstack: Orchestrating edge clouds using location-based focus of attention," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, Oct 2016, pp. 179–191.

[65] R. J. Hall, J. Auzins, J. Chapin, and B. Fell, "Scaling up a geographic addressing system," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, Nov 2013, pp. 143–149.

[66] D. L. Oppenheimer, J. Albrecht, D. Patterson, and A. Vahdat, "Scalable wide-area resource discovery," University of California, Berkeley, Tech. Rep. UCB/CSD-04-1334, 2004.

[67] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '04. New York, NY, USA: ACM, 2004, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/1015467.1015471

[68] A. Alsudais, Z. Huang, B. Balasubramanian, S. P. Narayanan, E. Keller, and K. Joshi, "Nodefinder: Scalable search over highly dynamic geo-distributed state," in *10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18)*. Boston, MA: USENIX Association, 2018. [Online]. Available: https://www.usenix.org/conference/hotcloud18/presentation/alsudais

[69] A. C.-C. Yao, "How to generate and exchange secrets (extended abstract)," in *27th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Oct. 1986, pp. 162–167.

[70] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, "Fairplay—a secure two-party computation system," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, ser. SSYM'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 20–20. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251375.1251395

[71] Y. Lindell and B. Pinkas, "A proof of security of Yao's protocol for two-party computation," *Journal of Cryptology*, vol. 22, no. 2, pp. 161–188, Apr. 2009.

[72] V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free XOR gates and applications," in *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, ser. Lecture Notes in Computer Science, L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, Eds., vol. 5126. Springer, Heidelberg, Jul. 2008, pp. 486–498.

[73] V. Kolesnikov, "Truly efficient string oblivious transfer using resettable tamper-proof tokens," in *TCC 2010: 7th Theory of Cryptography Conference*, ser. Lecture Notes in Computer Science, D. Micciancio, Ed., vol. 5978. Springer, Heidelberg, Feb. 2010, pp. 327–342.

[74] V. Kolesnikov and R. Kumaresan, "Improved OT extension for transferring short secrets," in *Advances in Cryptology – CRYPTO 2013, Part II*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds., vol. 8043. Springer, Heidelberg, Aug. 2013, pp. 54–70.

[75] V. Kolesnikov, P. Mohassel, and M. Rosulek, "FleXOR: Flexible garbling for XOR gates that beats free-XOR," in *Advances in Cryptology – CRYPTO 2014, Part II*, ser. Lecture Notes in Computer Science, J. A. Garay and R. Gennaro, Eds., vol. 8617. Springer, Heidelberg, Aug. 2014, pp. 440–457.

[76] S. Zahur, M. Rosulek, and D. Evans, "Two halves make a whole - reducing data transfer in garbled circuits using half gates," in *Advances in Cryptology – EUROCRYPT 2015, Part II*, ser. Lecture Notes in Computer Science, E. Oswald and M. Fischlin, Eds., vol. 9057. Springer, Heidelberg, Apr. 2015, pp. 220–250.

[77] V. Kolesnikov and A. J. Malozemoff, "Public verifiability in the covert model (almost) for free," in *Advances in Cryptology – ASIACRYPT 2015, Part II*, ser. Lecture Notes in Computer Science, T. Iwata and J. H. Cheon, Eds., vol. 9453. Springer, Heidelberg, Nov. / Dec. 2015, pp. 210–235.

[78] X. Wang, S. Ranellucci, and J. Katz, "Authenticated garbling and efficient maliciously secure two-party computation," in *ACM CCS 17: 24th Conference on Computer and Communications Security*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM Press, Oct. / Nov. 2017, pp. 21–37.

[79] V. Kolesnikov, R. Kumaresan, M. Rosulek, and N. Trieu, "Efficient batched oblivious PRF with applications to private set intersection," in *ACM CCS 16: 23rd Conference on Computer and Communications Security*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM Press, Oct. 2016, pp. 818–829.

[80] V. Kolesnikov, N. Matania, B. Pinkas, M. Rosulek, and N. Trieu, "Practical multi-party private set intersection from symmetric-key techniques," in *ACM CCS 17: 24th Conference on Computer and Communications Security*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM Press, Oct. / Nov. 2017, pp. 1257–1272.

[81] V. Pappas, F. Krell, B. Vo, V. Kolesnikov, T. Malkin, S. G. Choi, W. George, A. D. Keromytis, and S. Bellovin, “Blind seer: A scalable private DBMS,” in *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 359–374.

[82] B. A. Fisch, B. Vo, F. Krell, A. Kumarasubramanian, V. Kolesnikov, T. Malkin, and S. M. Bellovin, “Malicious-client security in blind seer: A scalable private DBMS,” in *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 395–410.

[83] RedHat, “What is multi-cloud?” https://www.redhat.com/en/topics/cloud-computing/what-is-multicloud.

[84] Forbes, “It’s A Multi-Cloud World, After All?” https://www.forbes.com/sites/adrianbridgwater/2018/09/21/its-a-multi-cloud-world-after-all/#462884ab3ae1.