The Cloud is Not Enough: Saving IoT from the Cloud

Ben Zhang, Nitesh Mor, John Kolb, Douglas S. Chan, Nikhil Goyal
Ken Lutz, Eric Allman, John Wawrzynek, Edward Lee, John Kubiatowicz
University of California, Berkeley

Abstract

The Internet of Things (IoT) represents a new class of applications that can benefit from cloud infrastructure. However, the current approach of directly connecting smart devices to the cloud has a number of disadvantages and is unlikely to keep up with either the growing speed of the IoT or the diverse needs of IoT applications.

In this paper we explore these disadvantages and argue that fundamental properties of the IoT prevent the current approach from scaling. What is missing is a well-architected system that extends the functionality of the cloud and provides seamless interplay among the heterogeneous components in the IoT space. We argue that raising the level of abstraction to a data-centric design—focused around the distribution, preservation and protection of information—provides a much better match to the IoT. We present early work on such a distributed platform, called the Global Data Plane (GDP), and discuss how it addresses the problems with the cloud-centric architecture.

1 Introduction

The market has seen an explosion in the number of smart devices. These latest devices offer rich interactivity by connecting to computing platforms and services [45, 50]. Featured by the growth of *Internet* connectivity and the augmentation of everyday *things*, this shift is commonly referred to as the Internet of Things (IoT) [25].

The IoT industry has benefited tremendously from the economic model of the cloud. With little investment in the infrastructure, even novice users can start collecting sensor data and streaming it back to the cloud [24]. Several IoT cloud platforms [5, 7, 19, 22] have gone further by offering easy-to-use APIs, data processing, visualization, and sample code for various hardware platforms.

On the "thing" side, hardware platforms such as Arduino [1], Raspberry Pi [15], and BeagleBone Black [4] have allowed easy and cheap prototyping for customized IoT applications. Companies [13, 17, 18] in this space offer complete solutions including hardware gateways, smartphone applications, web portals and cloud services.

With this blizzard of activity, the current trend seems to be that all peripherals, including sensors and actuators, communicate directly with the cloud and interact with each other through web services [51]. At first glance, this seems to be a natural architecture for IoT applications. However, several significant problems are revealed on closer inspection, including issues with privacy, security, scalability, latency, bandwidth, availability and durability control. While these problems are not new to typical web applications, they are exacerbated in the IoT space because of the fundamental differences between IoT and web services (see Sec. 3).

Our analysis suggests a need for a higher layer of abstraction for the IoT—one that more naturally fits the requirements of IoT applications while exploiting the underlying computing platforms that enable the IoT (like the cloud, the Fog [30] and gateways [62]). Our proposed abstraction is centered around data. It is focused on the transport, replication, preservation, and integrity of streams of data while enabling transparent optimization for locality and quality of service. We call the resulting infrastructure the Global Data Plane (GDP). Its foundation is the concept of a single-writer append-only log, coupled with location-independent routing, overlay multicast and higher level interfaces such as common access APIs (see Sec. 4).

The contributions of this paper are as follows:

- We present and summarize the state-of-the-art trends on IoT architecture design.
- We analyze the shortcomings of the existing architecture by explaining the fundamental differences between IoT applications and web services.
- We propose the design of a data-centric system for IoT applications.

Since this is an ongoing effort, we focus mainly on the design experience with GDP thus far.

2 Background

In this section, we review the state of the art in the distributed application space for IoT. The cloud has become the *de facto* foundation around which distributed applications are constructed—a trend that, while understandable, is not the best long-term approach (see Section 3).

2.1 Massive Adoption of Cloud Platforms

Over the last few years, cloud computing has shaped the software industry and made the development and deployment of web services easier than ever. Public cloud providers such as Amazon, Microsoft, Google, Rackspace offer pay-as-you-go services for the general public. Such a service model has reduced capital expenses, enabled elasticity for dynamic load adaption, and simplified resource management [24].

We have seen a huge recent trend of migrating computations and services to the cloud. For example, over two trillion objects were reported stored in Amazon S3 as of April 2013 [28]. Riding on this popularity, IoT application developers have adopted the cloud as a universal computation resource and a storage backend. This approach has been taken by industrial efforts (such as Carriots [5], GroveStreams [7], SAMI [22], Xively [19]), as well as academic research [44, 62].

2.2 Embedded Platforms

At the same time, we have seen a dizzying array of embedded platforms, from powerful computing units to low-power microcontrollers (see Table 1 for some examples). Below are three categories of embedded platforms:

1. Smartphones: Many companies (like Fitbit [6] or Automatic [3]) use smartphones as sensing devices as well as gateways to connect other low-power devices to the network. Researchers have explored how smartphones can be used for IoT including reusing discarded smartphones [32], writing new operating systems [10] and developing novel applications [47].

2. Mini PC: Ranging from the powerful Mac Mini and Intel Next Unit of Computing (NUC) to inexpensive Raspberry Pi and BeagleBone Black, these devices typically run various versions of Linux to simplify application deployment. Many companies [13, 17, 18] adopt these mini PCs as their gateway devices.

3. Microcontroller platforms: Examples include Arduino [1], mbed [2], and Particle [14]. This is an emerging category; there are new open platforms on crowdfunding websites [11], good ecosystems, great support and libraries (*e.g.* Adafruit Online Tutorials [12]) and novel applications/products [9].

[1]The original authors noted "Customer buyback price quoted by Sprint for a smartphone in good condition" [32].

Device	CPU Speed	Memory	Price
Intel NUC	1.3 GHz	16 GB	~$300
Typical Phones	2 GHz	2 GB	~$300
Discarded Phones[1]	1 GHz	512 MB	~$22
BeagleBone Black	1 GHz	512 MB	$55
Raspberry Pi	900 MHz	512 MB	$35
Arduino Uno	16 MHz	512 MB	~$22
mbed NXP LPC1768	96 MHz	32 KB	$10

Table 1: The world of IoT includes a wide spectrum of computing platforms.

2.3 IoT Application Status Quo

Many of today's IoT solutions arise by connecting embedded platforms to the cloud. For example, Bolt [44] provides data management for the Lab of Things (LoT) [31] and uses Amazon S3 or Azure for data storage.[2] Such direct connections often require an application-level gateway [62] to support low-power radios such as Z-Wave or Bluetooth Low Energy (BLE). Companies tend to provide their own gateways (such as Ninja Sphere [13], SmartThings Hub [17], Wink Hub [18]); and researchers adopt a similar approach (*e.g.* HomeHub for the LoT [31]). The fact that custom gateways are an integral part of IoT applications leads directly to "stovepipe" solutions or balkanization. Data and services from one company cannot be shared or utilized by devices from another company: connection protocols, data formats, and security mechanisms (when present) are proprietary and often undocumented.

To date, IoT applications fall into two general categories:

Ambient data collection and analytics: These applications involve sensors installed in buildings [36], at homes [46], in cities [16], and on humans themselves[3] [6, 59]. Normally, data is not immediately inspected and the collected data is later processed for analytics [48]. The trend of data collection is constantly growing and many researchers have predicted a new big-data problem [37, 63]. One thing to note is that many of the sensed data have serious privacy implications (for personal health, operational security, *etc.*).

Real-time applications with low-latency requirements: These applications could be reactive environments with humans in the loop [34]. An upper latency limit to avoid a notice by human participants is about 100 ms [53]. These applications could also be autonomous systems where humans are not involved (such as robots taking actions based on sensors). In this case, a tight control over latency is important for deterministic applications [39]. Tight latency requirements are often incompatible with the unpredictable performance of cloud-based analytics or controllers.

[2]This approach is for "efficiently sharing data across homes" [44].

[3]Often referred to as Quantified Self.

3 Pitfalls with Today's Approach to IoT

In this section, we argue why the current approach of connecting IoT devices directly to the cloud is incompatible with the evolving world of IoT applications. This incompatibility arises from the fundamental nature of IoT applications. Our reasoning is as follows:

1. Privacy and Security. Sensors implanted in our surrounding environment collect extremely sensitive information. In a recent talk given by Wadlow [60], he described the IoT as "hundreds of computers that are aware of me, can talk about me, and are out of my control." This is a strong call for intrinsic security and privacy. This need is echoed in many critical posts and talks (such as "Internet of Crappy Things" [38], "The Internet of Fails" [56]). The FTC's Technical Report [41] also emphasizes security in the IoT spaces. As a centralized resource out of users' control, the cloud resents an ever-present opportunity to violate privacy. Today, privacy has become a luxury [23], a situation that will be exacerbated in the IoT.

2. Scalability. By 2020, Cisco estimates 50 billion [40] devices will be connected to the cloud, while Gartner estimates 26 billion [52]. Scalability in the IoT spaces will be more challenging than web-scale or Internet-scale applications; the amount of data generated will easily exceed the reported trillion objects in Amazon S3 [28]. The bisection bandwidth requirements for a centralized cloud solution are staggering, especially since most data acquired by IoT devices can or should be processed locally and immediately discarded.

3. Modeling: peripheral devices are physical. Both sensors and actuators are physically present devices in our environment. Although sensor data can be collected and replicated (similar to virtualizing sensors [61]), the data is still generated from the edge of the network. Moreover, actuators cannot be virtualized and oftentimes the actuations cannot be rolled back. This is significantly different from the model of web services today.

4. Latency: The cloud model differs from reality. Application developers view the cloud as a component that interconnects the smart devices. However, from a network point of view, the cloud is on the edge of the network (see Fig. 1). Even simple IoT applications, such as those that turn on a fan in response to a rise of the local temperature, will experience unpredictable latencies from sensing, wireless transmission, gateway processing, Internet delivery, and cloud processing.

5. Bandwidth: upstream traffic dominates. Shipping data to the cloud incurs a significant amount of upstream traffic. Typical broadband networks have more downstream bandwidth than upstream bandwidth. IoT applications, however, generate data at the edges of the network, a pattern that will easily saturate the upstream link's bandwidth—especially at scale.

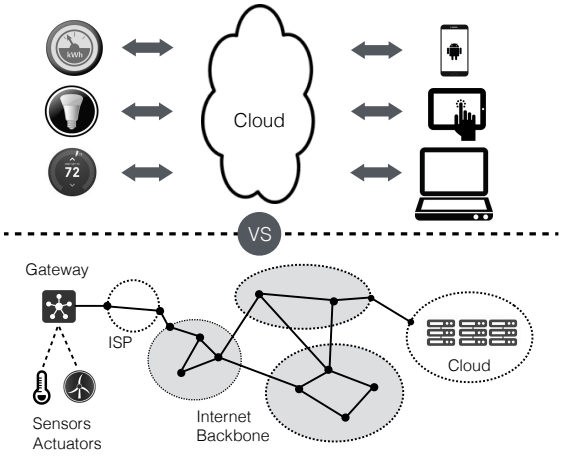

Figure 1: Although applications usually view the cloud as the center of all connected devices (*upper diagram*), in reality the cloud is usually on the edge of the Internet backbone, just like other devices (*lower diagram*).

For example, a single Dropcam requires "a high speed internet connection with at least 0.5 Mbps" to use its service [8]. Even simple sensors, such as energy meters, can benefit from a higher sampling rate (the motivation of 1 kHz energy data with ground-truth from the Ubicomplab at the University of Washington [43] and 15 kHz sampling of energy from MIT REDD Dataset [48]).

6. Quality of Service (QoS) Guarantees. Web users tolerate variable latency and occasional loss of web services. In contrast, the temporary unavailability of sensors or actuators within IoT applications will directly impact the physical world. While significant engineering effort has been put into improving the availability and latency profile of the cloud (allowing Service Level Agreements), such efforts are stymied by operator error, software bugs, DDoS attacks, and normal packet-to-packet variations from wide-area routing. Further, the Internet connection to people's homes is far from perfect. Over 10% of home networks in the developed world see connectivity interruptions more frequently than once every 10 days [42]; this situation is worse in developing countries.

7. Durability Management. Some sensor data is ephemeral, while other data should be durable against global disasters. For ephemeral data, there is no effective way of verifying the data has been completely destroyed because the cloud is out of the user's control. For durable data, regardless of the promised guarantees [21], the reliability of cloud storage remains a major concern and there is active research in this direction [29]. Moreover, whatever durability is achieved by the cloud, it is typically done so without concern for application-specific privacy or export rules. Note that control over durability is closely related to control in general: making sure that users retain the control and ownership over their data rather than providers.

4 A Data-Centric Proposal

The Global Data Plane (GDP) is a data-centric abstraction focused around the distribution, preservation, and protection of information. It supports the same application model as the cloud, while better matching the needs and characteristics of the IoT by utilizing heterogeneous computing platforms, such as small gateway devices, moderately powerful nodes in the environment and the cloud, in a distributed manner.

As shown in Fig. 2, the GDP interface provides a new "narrow waist" upon which applications are constructed. The basic foundation of the GDP is the secure, single-writer log. Logs in the GDP are lightweight, durable, and they support multiple simultaneous readers—either through random access (pull-based) or subscription (push-based). Logs have no fixed location but rather are migrated as necessary to meet locality, privacy, or QoS needs of applications.

Applications are built on top of the GDP by interconnecting log streams, rather than by addressing devices or services via IP. Each sensor or computational element of an IoT application has its own unique output log in the GDP and writes timestamped entries to this log. Actuators read from a unique input log. The GDP masks the heterogeneity of underlying communication paradigms, network/storage devices, and physical connections; and on top, it supports a wide variety of Common Access Application Program Interfaces (CAAPIs) for applications.

We detail a few key design decisions below:

1. Single-writer time-series logs: For each IoT device or application component that generates data, this data is represented as a log where the owner has the sole write permission. This model is based on our observation that peripherals are physical devices in our environment. We assume that devices have cryptographic keys for signing and encryption.[4] Logs are *append-only*; most data is *read-only* and can be securely replicated and validated through cryptographic hashes.

For each log, our current design exposes *append*, *read* and *subscribe* APIs. The single-writer model allows the following properties:

- *Flexibility:* The log interface is minimum but complete. Aggregations of logs or CAAPIs (discussed below) can be built by composition. In part (a) of Fig. 3, a new log is created by composing two existing ones and writing back to the GDP.
- *Access Control:* Since devices and services have associated public-key identities, each log has a single authorized writer. An *append* operation is permitted only when signed by the appropriate writer's key. For *read* operations, only those with an appropriate decryption key can decrypt the data, providing for a way to implement read-access control policies; a variety of more complex access control policies can be constructed through hierarchical key management or selected use of trusted environments.
- *Authenticity and integrity:* Since only signed *append* operations are allowed, accidental or malicious corruption of the log won't occur and substitution attacks are easily detected. A variety of traditional consistency problems are replaced with the simpler problem of finding the latest update.[5]
- *Encryption:* We envision that all data written to the log is encrypted with the encryption key held by the writer. A single writer with a single encryption key simplifies the key management challenges.
- *Durability and replication:* In contrast to the cloud where users rely on whatever durability the cloud providers offer, our model enables the choice of the level of durability and geographic span of replication on a per log basis. The log model also simplifies replica consistency as previously mentioned.

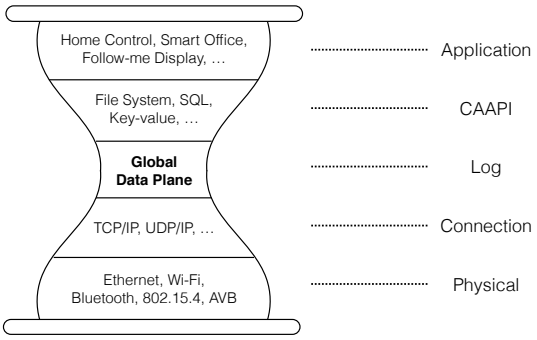

Figure 2: The Global Data Plane (GDP) operates above the network level and offers Common Access APIs (CAAPIs) to applications rather than raw packet routing. We argue that this abstraction is more appropriate for *both* IoT applications *and* the cloud.

2. Location-independent Routing: Logs must be physically stored in the infrastructure. As previously discussed, the current reliance of IoT on cloud storage provides few guarantees about the placement, latency of access, or durability of information. Instead, to embrace heterogeneous platforms and support a variety of storage policies, the GDP employs *location-independent routing* in a large, 256-bit address space. To meet the goal of flexible placement, controllable replication and easy migration, packets are routed through an overlay network that uses Distributed Hash Table (DHT) technology. DHT addresses the challenges of scalability [54, 58, 65] with the sacrifice of an increased number of overlay hops. GDP optimizes latency through log migration (see Fig. 3(d)) and dynamic changes to the routing topology.

[4] In case of extremely low power sensors, the cryptographic operations could be performed by a more powerful gateway device.

[5] A single writer can track a hash of the latest update in non-volatile memory. More sophisticated multi-writer services can utilize Byzantine agreement to serve up the latest write to interested parties.

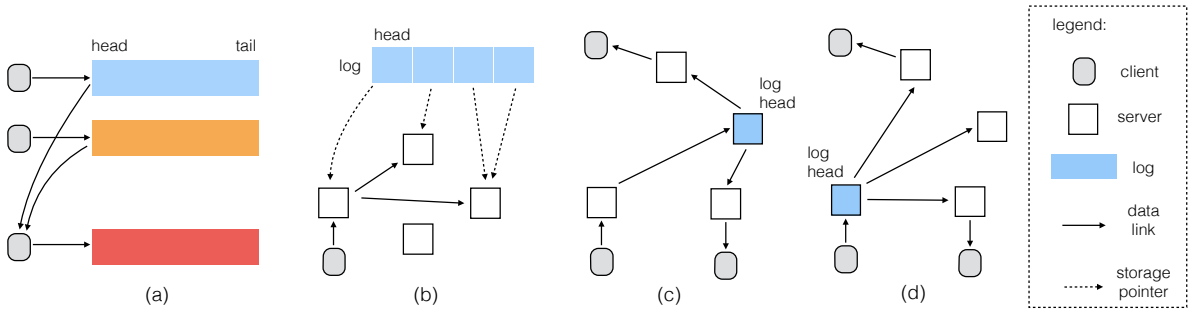

Figure 3: The GDP design illustrated: (a) single-writer logs are appended to the head and compositions are achieved by subscription; (b) logs are split into chunks and stored in a distributed fashion; (c) overlay multicast trees are constructed when there are multiple subscribers; (d) location-independent routing enables log migration for optimizing performance.

Logs are named with a 256-bit identifier which may be derived from a cryptographic hash of the owner's public key and metadata. Following a variety of placement and replication policies,[6] the GDP places logs within the infrastructure and advertises the location of these logs to the underlying routing layer. Such placement and replication policies can optimize for latency, QoS, privacy, durability, and so forth. Internally, logs are further split into chunks, and each chunk can be distributed for durability [49] and performance [44] (see Fig. 3(b)).

3. Pub/Sub and multicast tree: The publish/subscribe pattern has been shown to support a wide variety of fundamental communication services (for mobility, multicast, anycast [57]). This fits nicely with our log abstraction and can support building interactive applications. To alleviate the growth of sensor data bandwidth, when multiple subscribers exist, multicast trees can be built on top of the overlay network using techniques proposed earlier [27, 65] so that effective bandwidth is reduced [33] (see Fig. 3(c)).

4. Common Access API (CAAPI): Although the single-writer log abstraction shelters developers from low-level machine and communication primitives, many applications are likely to need more common APIs or data structures [26]. In fact, logs are sufficient to implement any convenient, mutable data storage repository. Thus, Fig. 2 shows a CAAPI layer on top of the GDP. A CAAPI can provide key-value store, file system or database operations. Since logs serve as the ground truth, the benefit of consistency, durability, scalability and availability are carried over to CAAPIs for free. However CAAPIs may need to replay the logs if the service fails; in this case, checkpointing can be employed to avoid expensive log replay.

Our design for the GDP is not yet bullet-proof and our initial implementation has not withstood the test of wide-scale deployment. Nonetheless, we believe that the core concepts of GDP overcome the pitfalls mentioned in Sec. 3 in the following way: the single-writer, append-only log models sensor data more accurately; integrity and authentication by design provides better privacy and security; the distributed nature with peer-to-peer technology makes scalability possible; explicit separation of policy from mechanism enables better control on level of durability for end users; and finally, latency, bandwidth and QoS guarantees are enabled by the integration of the cloud and the local infrastructure.

[6]How to specify a policy about where logs are placed is out of the scope of this paper. We leave it as a future work.

5 Related Work

Other efforts exist to address the challenges of IoT. Cisco's Fog Computing [30] provides computing resources closer to the edge of the network. We believe that our arguments strengthen the need for fog-like computing platforms and our proposed GDP architecture can leverage such resources. Also relevant are systems such as EdgeComputing from Akamai [35], Intel's Intelligent Edge [20], and Microsoft's Cloudlet [55]. The role of servers in these architectures seems to emphasize on being intelligent gateways or proxies for data flowing into and from the cloud. Support for an entirely decentralized data storage and delivery platform is apparently absent.

Our data-centric design hails from Oceanstore [49] and shares a number of goals with Named Data Networking [64], but our focus on the IoT application space leads to a number of important design differences. A few design decisions are similar to Bolt [44]: single-writer time-series data, chunking for performance, efficient data sharing, policy-driven storage and data confidentiality/integrity. However, Bolt takes the cloud approach where the pitfalls in Sec. 3 are unavoidable.

6 Acknowledgments

This work was supported in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

[1] Arduino. http://www.arduino.cc/.

[2] ARM mbed. https://mbed.org/.

[3] Automatic. https://www.automatic.com/.

[4] BeagleBone Board. http://beagleboard.org/bone.

[5] Carriots. https://www.carriots.com/.

[6] Fitbit. http://www.fitbit.com/.

[7] GroveStreams. https://www.grovestreams.com/.

[8] How much bandwidth does Dropcam use? http://support.dropcam.com/entries/21438818-How-much-bandwidth-does-Dropcam-use.

[9] iotlist: Discover the Internet of Things. http://iotlist.co/category/Kickstarter.

[10] JanOS: Turn your phone into an IoT board. http://janos.io/.

[11] KickStarter. https://www.kickstarter.com/.

[12] Learning at Adafruit Industries, Unique & fun DIY electronics and kits. https://learn.adafruit.com.

[13] Ninja Blocks. https://ninjablocks.com/.

[14] Particle. https://www.particle.io/.

[15] Raspberry Pi. http://www.raspberrypi.org/.

[16] SFPark. http://sfpark.org/.

[17] SmartThings. http://www.smartthings.com/.

[18] Wink. http://www.wink.com/.

[19] Xively. https://xively.com/.

[20] Intel Architecture at the Edge for Greater Flexibility and Scalability. http://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/communications-intel-architecture-brief.pdf, 2011.

[21] Amazon S3 Durability. http://aws.amazon.com/s3/faqs/#Data_Protection/, 2015.

[22] Samsung SAMI: A data exchange platform that defines a new paradigm. https://developer.samsungsami.io/, 2015.

[23] ANGWIN, J. Has Privacy Become a Luxury Good? http://www.nytimes.com/2014/03/04/opinion/has-privacy-become-a-luxury-good.html, 2014.

[24] ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A. D., KATZ, R., KONWINSKI, A., LEE, G., PATTERSON, D., RABKIN, A., STOICA, I., ET AL. A view of cloud computing. *Communications of the ACM 53*, 4 (2010), 50–58.

[25] ATZORI, L., IERA, A., AND MORABITO, G. The internet of things: A survey. *Computer networks 54*, 15 (2010), 2787–2805.

[26] BALAKRISHNAN, M., MALKHI, D., WOBBER, T., WU, M., PRABHAKARAN, V., WEI, M., DAVIS, J. D., RAO, S., ZOU, T., AND ZUCK, A. Tango: Distributed data structures over a shared log. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* (2013), ACM, pp. 325–340.

[27] BALLARDIE, T., FRANCIS, P., AND CROWCROFT, J. Core based trees (CBT). In *ACM SIGCOMM Computer Communication Review* (1993), vol. 23, ACM, pp. 85–95.

[28] BARR, J. Amazon S3 Two Trillion Objects, 1.1 Million Requests / Second. https://aws.amazon.com/blogs/aws/amazon-s3-two-trillion-objects-11-million-requests-second/, 2013.

[29] BESSANI, A., CORREIA, M., QUARESMA, B., ANDRÉ, F., AND SOUSA, P. DepSky: dependable and secure storage in a cloud-of-clouds. *ACM Transactions on Storage (TOS) 9*, 4 (2013), 12.

[30] BONOMI, F., MILITO, R., ZHU, J., AND ADDEPALLI, S. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (2012), ACM, pp. 13–16.

[31] BRUSH, A., FILIPPOV, E., HUANG, D., JUNG, J., MAHAJAN, R., MARTINEZ, F., MAZHAR, K., PHANISHAYEE, A., SAMUEL, A., SCOTT, J., ET AL. Lab of things: a platform for conducting studies with connected devices in multiple homes. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (2013), ACM, pp. 35–38.

[32] CHALLEN, G., HASELEY, S., MAITI, A., NANDUGUDI, A., PRASAD, G., PURI, M., AND WANG, J. The mote is dead: long live the discarded smartphone! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications* (2014), ACM, p. 5.

[33] CHU, Y.-H., RAO, S. G., SESHAN, S., AND ZHANG, H. A case for end system multicast. *Selected Areas in Communications, IEEE Journal on 20*, 8 (2002), 1456–1471.

[34] COOPERSTOCK, J. R., FELS, S. S., BUXTON, W., AND SMITH, K. C. Reactive environments. *Communications of the ACM 40*, 9 (1997), 65–73.

[35] DAVIS, A., PARIKH, J., AND WEIHL, W. E. Edgecomputing: extending enterprise applications to the edge of the internet. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (2004), ACM, pp. 180–187.

[36] DAWSON-HAGGERTY, S., JIANG, X., TOLLE, G., ORTIZ, J., AND CULLER, D. sMAP: a simple measurement and actuation profile for physical information. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems* (2010), ACM, pp. 197–210.

[37] DIAZ, M., JUAN, G., AND OIKAWA LUCAS, A. R. Big Data on the Internet of Things. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (2012), pp. 978–0.

[38] DROZHZHIN, A. Internet of Crappy Things. http://blog.kaspersky.com/internet-of-crappy-things/, 2015.

[39] EIDSON, J. C., LEE, E. A., MATIC, S., SESHIA, S. A., AND ZOU, J. Distributed real-time software for cyber–physical systems. *Proceedings of the IEEE 100*, 1 (2012), 45–59.

[40] EVANS, D. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper 1* (2011).

[41] FTC. Internet of Things, Privacy & Security in a Connected World. http://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf, 2015.

[42] GROVER, S., PARK, M. S., SUNDARESAN, S., BURNETT, S., KIM, H., RAVI, B., AND FEAMSTER, N. Peeking behind the NAT: an empirical study of home networks. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 377–390.

[43] GUPTA, S., LARSON, E., AND PATEL, S. Household Energy Dataset. http://ubicomplab.cs.washington.edu/projects/datasets, 2015.

[44] GUPTA, T., SINGH, R. P., AND MAHAJAN, A. P. J. J. R. Bolt: Data management for connected homes. In *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)* (2014), pp. 243–256.

[45] HARTMANN, B., AND WRIGHT, P. K. Designing bespoke interactive devices. *Computer*, 8 (2013), 85–89.

[46] HNAT, T. W., SRINIVASAN, V., LU, J., SOOKOOR, T. I., DAWSON, R., STANKOVIC, J., AND WHITEHOUSE, K. The hitchhiker's guide to successful residential sensing deployments. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (2011), ACM, pp. 232–245.

[47] HONG, J.-H., MARGINES, B., AND DEY, A. K. A smartphone-based sensing platform to model aggressive driving behaviors. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), ACM, pp. 4047–4056.

[48] KOLTER, J. Z., AND JOHNSON, M. J. REDD: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA* (2011), vol. 25, Citeseer, pp. 59–62.

[49] KUBIATOWICZ, J., BINDEL, D., CHEN, Y., CZERWINSKI, S., EATON, P., GEELS, D., GUMMADI, R., RHEA, S., WEATHERSPOON, H., WEIMER, W., ET AL. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices 35*, 11 (2000), 190–201.

[50] KUNIAVSKY, M. *Smart Things: Ubiquitous Computing User Experience Design: Ubiquitous Computing User Experience Design*. Elsevier, 2010.

[51] LEE, E. A., RABAEY, J., BLAAUW, D., DUTTA, P., FU, K., GUESTRIN, C., HARTMANN, B., JAFARI, R., JONES, D., KUBIATOWICZ, J., ET AL. The Swarm at the Edge of the Cloud.

[52] MIDDLETON, P., KJELDSEN, P., AND TULLY, J. Forecast: The internet of things, worldwide, 2013. *Gartner Research* (2013).

[53] NIELSEN, J. *Usability engineering*. Elsevier, 1994.

[54] ROWSTRON, A., AND DRUSCHEL, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware 2001* (2001), Springer, pp. 329–350.

[55] SATYANARAYANAN, M., BAHL, P., CACERES, R., AND DAVIES, N. The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE 8*, 4 (2009), 14–23.

[56] STANISLAV, M., AND LANIER, Z. The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right. `https://www.defcon.org/html/defcon-22/dc-22-speakers.html`, 2014.

[57] STOICA, I., ADKINS, D., ZHUANG, S., SHENKER, S., AND SURANA, S. Internet Indirection Infrastructure. In *ACM SIGCOMM Computer Communication Review* (2002), vol. 32, ACM, pp. 73–86.

[58] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review 31*, 4 (2001), 149–160.

[59] SWAN, M. The quantified self: fundamental disruption in big data science and biological discovery. *Big Data 1*, 2 (2013), 85–99.

[60] WADLOW, T. The Questions are the Same, but the Answers are Always Changing. `https://swarmlab.eecs.berkeley.edu/events/2014/11/18/5165/questions-are-same-answers-are-always-changing`, 2015.

[61] YURIYAMA, M., AND KUSHIDA, T. Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on* (2010), IEEE, pp. 1–8.

[62] ZACHARIAH, T., KLUGMAN, N., CAMPBELL, B., ADKINS, J., JACKSON, N., AND DUTTA, P. The Internet of Things Has a Gateway Problem. In *HotMobile'15* (2015), ACM, pp. 27–32.

[63] ZASLAVSKY, A., PERERA, C., AND GEORGAKOPOULOS, D. Sensing as a service and big data. *arXiv preprint arXiv:1301.0159* (2013).

[64] ZHANG, L., AFANASYEV, A., BURKE, J., JACOBSON, V., CROWLEY, P., PAPADOPOULOS, C., WANG, L., ZHANG, B., ET AL. Named Data Networking. *ACM SIGCOMM Computer Communication Review 44*, 3 (2014), 66–73.

[65] ZHAO, B. Y., KUBIATOWICZ, J., JOSEPH, A. D., ET AL. Tapestry: An infrastructure for fault-tolerant wide-area location and routing.