The Emergence of Edge Computing

Mahadev Satyanarayanan, Carnegie Mellon University

Industry investment and research interest in edge computing, in which computing and storage nodes are placed at the Internet's edge in close proximity to mobile devices or sensors, have grown dramatically in recent years. This emerging technology promises to deliver highly responsive cloud services for mobile computing, scalability and privacy–policy enforcement for the Internet of Things, and the ability to mask transient cloud outages.

Cloud computing, which has dominated IT discourse in the past decade, has a twofold value proposition. First, centralization exploits economies of scale to lower the marginal cost of system administration and operations. Second, organizations can avoid the capital expenditure of creating a datacenter by consuming computing resources over the Internet from a large service provider. These considerations have led to the consolidation of computing capacity into multiple large datacenters spread across the globe. The proven economic benefits of cloud computing make it likely to remain a permanent feature of the future computing landscape.

However, the forces driving centralization are not the only ones at work. Nascent technologies and applications for mobile computing and the Internet of Things (IoT) are driving computing toward dispersion. *Edge computing* is a new paradigm in which substantial computing and storage resources—variously referred to as cloudlets,[1] micro datacenters, or fog nodes[2]—are placed at the Internet's edge in close proximity to mobile devices or sensors.

Industry investment and research interest in edge computing have grown dramatically in recent years. Nokia and IBM jointly introduced the Radio Applications Cloud Server (RACS), an edge computing platform for 4G/LTE networks, in early 2013.[3] The following year, a mobile edge computing standardization effort began under the auspices of the European Telecommunications Standards Institute (ETSI).[4] The Open Edge Computing initiative (OEC; openedgecomputing.org) was launched in June 2015 by Vodafone, Intel, and Huawei in partnership with Carnegie Mellon University (CMU) and expanded a year later to include Verizon, Deutsche Telekom, T-Mobile, Nokia, and Crown Castle. This collaboration

includes creation of a Living Edge Lab in Pittsburgh, Pennsylvania, to gain hands-on experience with a live deployment of proof-of-concept cloudlet-based applications. Organized by the telecom industry, the first Mobile Edge Computing Congress (tmt.knect365.com /mobile-edge-computing) convened in London in September 2015 and again in Munich a year later. The Open Fog Consortium (www.openfogconsortium .org) was created by Cisco, Microsoft, Intel, Dell, and ARM in partnership with Princeton University in November 2015, and has since expanded to include many other companies. The First IEEE/ ACM Symposium on Edge Computing (conferences.computer.org/SEC) was held in October 2016 in Washington, DC.

These developments raise several questions: why has edge computing emerged, what new capabilities does it enable, and where is it headed?

ORIGIN AND BACKGROUND

The roots of edge computing reach back to the late 1990s, when Akamai introduced content delivery networks (CDNs) to accelerate web performance.[5] A CDN uses nodes at the edge close to users to prefetch and cache web content. These edge nodes can also perform some content customization, such as adding location-relevant advertising. CDNs are especially valuable for video content, because the bandwidth savings from caching can be substantial.

Edge computing generalizes and extends the CDN concept by leveraging cloud computing infrastructure. As with CDNs, the proximity of cloudlets to end users is crucial. However, instead of being limited to caching web content, a cloudlet can run arbitrary code just as in cloud computing. This code is typically encapsulated in a virtual machine (VM) or a lighter-weight container for isolation, safety, resource management, and metering.

In 1997, Brian Noble and his colleagues first demonstrated edge computing's potential value to mobile computing.[6] They showed how speech recognition could be implemented with acceptable performance on a resource-limited mobile device by offloading computation to a nearby server. Two years later, Jason Flinn and I extended this approach to improve battery life.[7] In a 2001 article that generalized these concepts, I introduced the term *cyber foraging* for the amplification of a mobile device's computing capabilities by leveraging nearby infrastructure.[8]

Cloud computing's emergence in the mid-2000s led to the cloud becoming the most obvious infrastructure to leverage from a mobile device. Today, Apple's Siri and Google's speech-recognition services both offload computation to the cloud. Unfortunately, consolidation implies large average separation between a mobile device and its optimal cloud datacenter. Ang Li and his colleagues reported that the average round-trip time from 260 global vantage points to their optimal Amazon Elastic Compute Cloud (EC2) instances is 74 ms.[9] To this must be added the latency of a wireless first hop. In terms of jitter, the variance inherent in a multihop network must be included.

Clearly, reliance on a cloud datacenter is not advisable for applications that require end-to-end delays to be tightly controlled to less than a few tens of milliseconds. As will be discussed later, tight control of latency is necessary

> [**USING PERSISTENT CACHING SIMPLIFIES THE MANAGEMENT OF CLOUDLETS DESPITE THEIR PHYSICAL DISPERSAL AT THE INTERNET EDGE.**]

for emerging applications such as augmented reality (AR).

These observations about end-to-end latency and cloud computing were first articulated in a 2009 article I coauthored with Paramvir Bahl, Rámon Cáceres, and Nigel Davies that laid the conceptual foundation for edge computing.[1] We advocated a two-level architecture: the first level is today's unmodified cloud infrastructure; the second level consists of dispersed elements called cloudlets with state cached from the first level. Using persistent caching instead of hard state simplifies the management of cloudlets despite their physical dispersal at the Internet edge. The cloudlet concept can, of course, be expanded to a multilevel cloudlet hierarchy.

In 2012, Flavio Bonomi and his colleagues introduced the term *fog computing* to refer to this dispersed cloud infrastructure.[2] However, their motivation for decentralization is IoT infrastructure scalability rather than mobile applications' interactive performance. The researchers envision a multilevel hierarchy of fog nodes
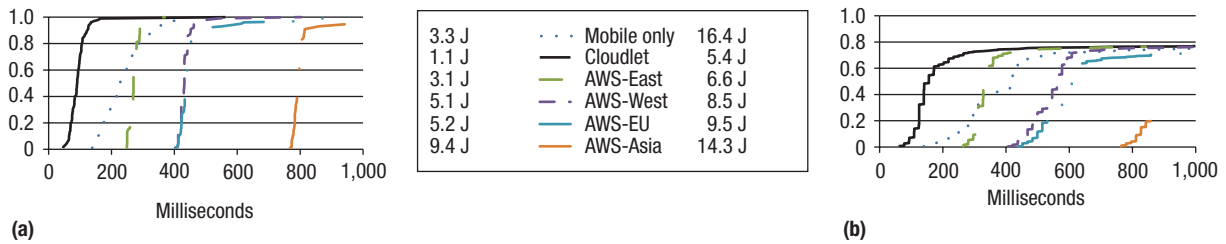
FIGURE 1. Response time distribution and per-operation energy cost of an (a) augmented reality and (b) face recognition application on a mobile device, in which an image from the device is transmitted over a Wi-Fi first hop to a cloudlet or an Amazon Web Services (AWS) datacenter. The ideal is best approximated by a cloudlet, demonstrating the importance of low-latency offload services. Figure adapted from K. Ha et al., "The Impact of Mobile Multimedia Applications on Data Center Consolidation," *Proc. 2013 IEEE Int'l Conf. Cloud Eng.* (IC2E 13), 2013, pp. 166–176.

stretching from the cloud to IoT edge devices.

WHY PROXIMITY MATTERS

As we explore new applications and use cases for both mobile computing and the IoT, the virtues of proximity are becoming increasingly apparent. In the physical world, the importance of proximity has never been in doubt. The old axiom about the three top determinants of real estate value being "location, location, and location" captures this observation well. In the cyber world, the seamless connectivity offered by the Internet has lulled us into a false sense of disregard for physical proximity. Because logical network proximity is entirely characterized by low latency, low jitter, and high bandwidth, the question "How close is physically close enough?" cannot be answered in the abstract. It is dependent on factors such as the networking technologies used, network contention, application characteristics, and user tolerance for poor interactive response.

Physical proximity affects end-to-end latency, economically viable bandwidth, establishment of trust, and survivability. With sufficient effort and resource investment, the lack of proximity can be partially masked. For example, a direct fiber connection can achieve low latency and high bandwidth between distant points. However, there are limits to this approach. The speed of light is an obvious physical limit on latency. The

need to use a multihop networking strategy to cover a large geographic area with many access points imposes an economic limit on both latency and bandwidth. Each hop introduces queuing and routing delay, as well as buffer bloat.[10]

The proximity of cloudlets helps in at least four distinct ways:

> *Highly responsive cloud services.* A cloudlet's physical proximity to a mobile device makes it easier to achieve low end-to-end latency, high bandwidth, and low jitter to services located on the cloudlet. This is valuable for applications such as AR and virtual reality that offload computation to the cloudlet.

> *Scalability via edge analytics.* The cumulative ingress bandwidth demand into the cloud from a large collection of high-bandwidth IoT sensors, such as video cameras, is considerably lower if the raw data is analyzed on cloudlets. Only the (much smaller) extracted information and metadata must be transmitted to the cloud.

> *Privacy-policy enforcement.* By serving as the first point of contact in the infrastructure for IoT sensor data, a cloudlet can enforce the privacy policies of its owner prior to release of the data to the cloud.

> *Masking cloud outages.* If a cloud service becomes unavailable due

to network failure, cloud failure, or a denial-of-service attack, a fallback service on a nearby cloudlet can temporarily mask the failure.

I now discuss each of these advantages in detail.

HIGHLY RESPONSIVE CLOUD SERVICES

Humans are acutely sensitive to delays in the critical path of interaction, and their performance on cognitive tasks is remarkably fast and accurate.[11] For example, under normal lighting conditions, face recognition takes 370–620 ms, depending on familiarity. Speech recognition takes 300–450 ms for short phrases, and it requires only 4 ms to tell that a sound is a human voice. VR applications that use head-tracked systems require latencies of less than 16 ms to achieve perceptual stability. End-to-end latency of a few tens of milliseconds is a safe but achievable goal.

Figure 1 illustrates the importance of cloudlets for low-latency offload services. The graphs show the cumulative distribution of measured response times for an AR and a face recognition application on a mobile device.[12] An image from the mobile device, which is located in Pittsburgh, is transmitted over a Wi-Fi first hop to a cloudlet or to an Amazon Web Services (AWS) datacenter. The image is processed at the destination by computer vision code executing

within a VM. For AR, buildings in the image are recognized and labels corresponding to their identities are transmitted back to the mobile device. For face recognition, the identity of the person is returned.

The ideal curve in Figure 1 would be a step function that jumps to 1.0 at the origin. As the figure shows, the ideal is best approximated by a cloudlet. End-to-end network latency impedes performance, as indicated by the worsening response-time curves corresponding to more distant AWS locations. Increasing response time also increases per-operation energy consumption on the mobile device. This value is indicated beside the corresponding label in the figure legend. For example, the device consumes 1.1 J on average to perform an AR operation on the cloudlet, but 3.1 J, 5.1 J, and so on when performing it on AWS-East, AWS-West, and so on. Similar results can be expected with any offload service that is concentrated in a few large datacenters.

The label "mobile only" in the figure corresponds to a case where no offloading is performed and the computer vision code is run on the mobile device. In spite of avoiding the energy and performance cost of Wi-Fi communication, this option is slower than using the cloudlet. Offloading is clearly important for these applications.

Cloudlets are a disruptive technology that brings energy-rich high-end computing within one wireless hop of mobile devices, thereby enabling new applications that are both computation-intensive and latency-sensitive. A prime example is *wearable cognitive assistance*,[11] which combines a device like Google Glass with cloudlet-based processing to guide users through a complex task.

As with a GPS system, the user hears a synthesized voice describing what to do next and sees visual cues in the Glass display. The system catches errors immediately and corrects the user before they cascade. The final report of the 2013 National Science Foundation

Workshop on Future Directions in Wireless Networking characterized this new genre of applications as "astonishingly transformative."[13] In ongoing work at CMU,[14] we have built cognitive assistance applications for the seven tasks summarized in Table 1. Videos of some of these applications are available at goo.gl/02m0nL.

On the cloudlet, the workflow of these applications consists of two phases. In the first phase, the sensor inputs are analyzed to extract a symbolic representation of task progress (fourth column of Table 1). This is an idealized representation of the input sensor values relative to the task, and excludes all irrelevant detail. This phase must be tolerant of considerable real-world variability—for example, different lighting levels, light sources, viewer's positions with respect to the task artifacts, task-unrelated clutter in the background, and so on. One can view the extraction of a symbolic representation as a task-specific "analog-to-digital" conversion: the enormous state space of sensor values is simplified to a much
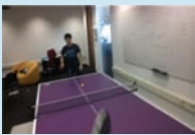
smaller task-specific state space. The second phase of each task workflow operates solely on the symbolic representation. Comparing the symbolic representation to the expected task state generates user guidance for the next step (last column of Table 1). The

video guidance is shown on the Glass display, and audio guidance is given using the Android text-to-speech API.

SCALABILITY THROUGH EDGE ANALYTICS

Independent of latency considerations, cloudlets can also reduce ingress bandwidth into the cloud. For example, consider an application in which many colocated users are continuously transmitting video from their smartphone to the cloud for content analysis. The cumulative data rate for even a small fraction of users in a modest-size city would saturate its metropolitan area network: 12,000 users transmitting 1080p video would require a link of 100 gigabits per second; a million users would require a link of 8.5 terabits per second.

Figure 2 shows how cloudlets can solve this problem. In the proposed GigaSight framework,[15] video from a mobile device only travels as far as a nearby cloudlet. The cloudlet runs computer vision analytics in near real time and only sends the results (content tags, recognized faces, and

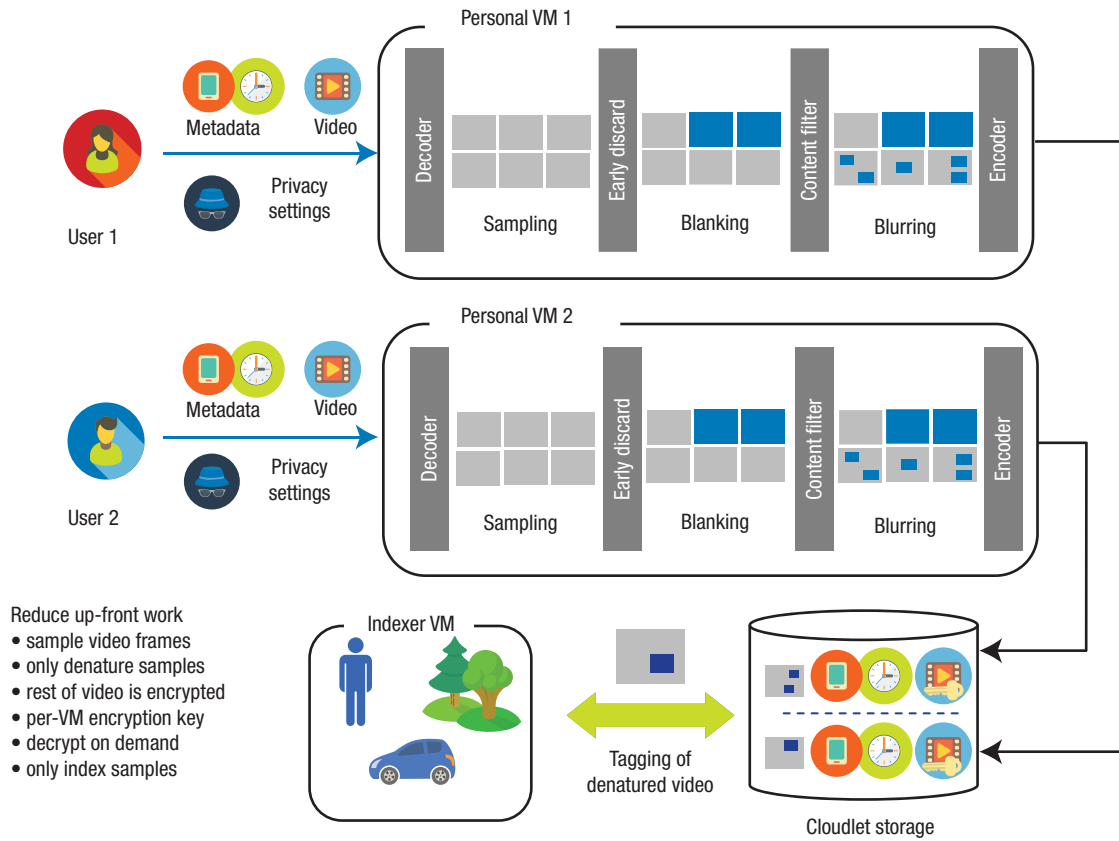|		**TABLE 1.** Example wearable cognitive assistance applications.		
App name	**Example input video frame**	**App description**	**Symbolic representation**	**Example guidance**
Face		Jogs user's memory of a familiar face whose name cannot be recalled. Detects and extracts a tightly cropped image of each face, then applies popular open source face recognizer OpenFace (cmusatyalab.github.io/openface), which is based on a deep neural network (DNN) algorithm. Whispers name of person. Can be used in combination with mood detection algorithms to offer conversational hints.	ASCII text of name	Whispers "Barack Obama"
Pool		Helps novice pool player aim correctly. Gives continuous visual feedback (left arrow, right arrow, or thumbs up) as user turns cue stick. Correct shot angle is calculated based on widely used fractional aiming system. Uses color, line, contour, and shape detection. Symbolic representation describes positions of cue ball, object ball, target pocket, and top and bottom of cue stick.	<Pocket, object ball, cue ball, cue top, cue bottom>	
Ping-Pong		Tells novice player to hit ball to left or right, depending on which is more likely to beat opponent. Uses color, line, and optical-flow-based motion detection to detect ball, table, and opponent. Symbolic representation is a 3-tuple: in rally or not, opponent position, ball position. Whispers "left" or "right."	<InRally, ball position, opponent position>	Whispers "Left"
Workout		Guides correct user form in exercise actions like sit-ups and push-ups, and counts out repetitions. Uses volumetric template matching on a 10- to 15-frame video segment to classify poorly performed repetitions as distinct types of exercise (for example, "bad push-up"). Uses smartphone on floor for third-person viewpoint.	<Action count>	Says "8"
Lego		Guides user in assembling 2D Lego models. Analyzes each video frame in three steps: (1) finds board using its distinctive color and black dot pattern, (2) locates Lego bricks on board using edge and color detection, and (3) assigns brick color using weighted majority voting within each block. Symbolic representation is matrix showing color for each brick.	[[0, 2, 1, 1], [0, 2, 1, 6], [2, 2, 2, 2]]	Says "Find a 1 × 3 green piece and put it on top"
Draw		Helps user to sketch better. Builds on third-party app originally designed to input sketches from pen tablets and output corrective guidance on desktop screen. Our implementation preserves back-end logic. New Google Glass–based front end allows use of any drawing surface and instrument and displays guidance on Glass. Displays error alignment in sketch.		
Sandwich		Helps cooking novice prepare sandwiches according to a recipe. Because real food is perishable, we use realistic plastic toy food as ingredients. Object detection uses a region proposal and DNN approach. Implementation is on top of Caffe (caffe.berkeleyvision.org) and Dlib (dlib.net). Transfer learning saves time in labeling and training.	Object: "Lettuce on top of ham and bread"	Says "Now put a piece of bread on the lettuce"

FIGURE 2. GigaSight framework. A cloudlet performs computer vision analytics on video from mobile devices in near real time and only sends the results along with metadata to the cloud, sharply reducing ingress bandwidth into the cloud. VM: Virtual machine.

so on) along with metadata (owner, capture location, timestamp, and so on) to the cloud. This can reduce ingress bandwidth into the cloud by three to six orders of magnitude. GigaSight also shows how tags and metadata in the cloud can guide deeper and more customized searches of the content of a video segment during its (finite) retention period on a cloudlet.

A video camera is only one example of a high-data-rate sensor in the IoT. Another example is a modern aircraft, which can generate nearly half a terabyte of sensor data during a flight. Real-time analysis of this data on a cloudlet in the aircraft could generate timely guidance for preventive maintenance, fuel economy, and other benefits.

Cloudlets' latency and bandwidth advantages are especially relevant in the context of automobiles, to complement vehicle-to-vehicle (V2V)

approaches being explored for real-time control and accident avoidance. For the foreseeable future, cloud connectivity from a moving automobile will at best be 3G or 4G/LTE. An important question is whether cloudlets should be in automobiles or part of the telco infrastructure (perhaps one cloudlet connected via fiber links to multiple cell towers in an area). Both alternatives have value.

An application such as a multiplayer video game for automobile passengers is best hosted on the vehicle's cloudlet. The cloudlet could also perform real-time analytics of high-data-rate sensor streams from the engine and other sources to alert the driver to imminent failure or the need for preventive maintenance. In addition, this information could be transmitted to the cloud for integration into the vehicle manufacturer's database. Fine-grain analysis of such anomaly data

might reveal model-specific defects that could be corrected in a timely manner.

For other automotive applications, such as collaborative real-time avoidance of road hazards, the telco cloudlet is the optimal hosting site. For example, if a vehicle hits a pothole or swerves to avoid a fallen tree branch, the hazard's coordinates can be rapidly shared within the telco cloudlet and then used for many hours by other automobiles to proactively cope with the hazard (for example, by warning drivers to make an early lane change).

PRIVACY POLICY ENFORCEMENT

Cloudlets could help address a vexing problem—namely, growing concerns over data privacy arising from IoT system overcentralization. Increasingly reluctant to release raw sensor data to an IoT cloud hub, users
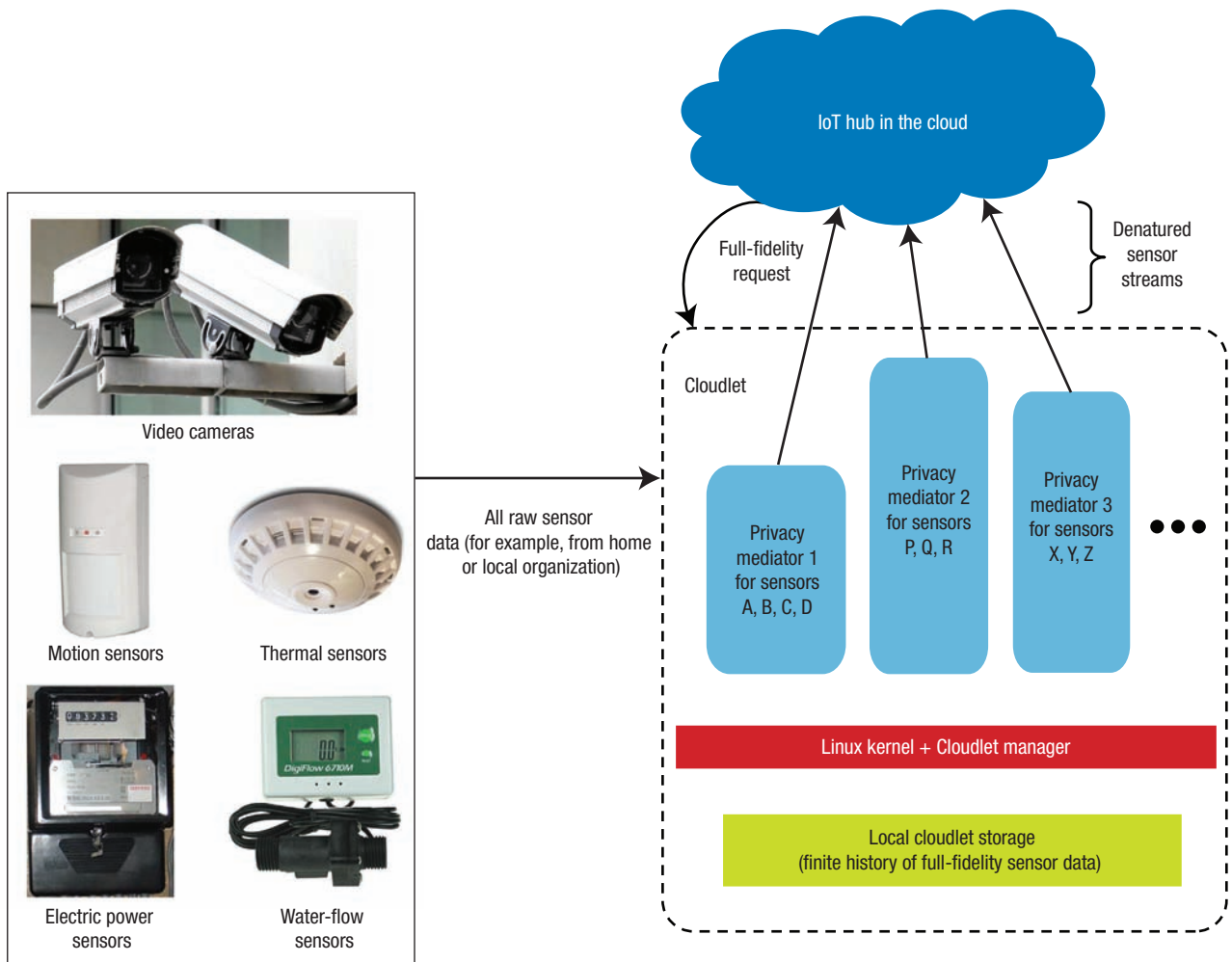
FIGURE 3. Internet of Things (IoT) privacy architecture. Software modules called privacy mediators execute on a cloudlet within a sensor owner's trust domain to perform denaturing and privacy–policy enforcement on the sensor streams.

and organizations desire finer-grain control over release of that data. For example, a user should be able to delete or denature a subset of sensor data he or she deems sensitive. From the end user's per-spective, denatured sensor data is safe to release to the outside world: faces in images can be blurred, sensor readings can be coarsely aggregated or omitted at certain times of day or night, and so on. Today's IoT architectures, in which data is transmitted directly from sensors to a cloud hub, make such fine-grain control impossible.

Nigel Davies and his colleagues[16] propose an IoT privacy architecture (see Figure 3) that leverages a cloudlet within a sensor owner's trust domain. This cloudlet is the first point of infrastructure contact for the sensor streams. Trusted software modules called *privacy mediators* execute on the cloudlet to perform denaturing and privacy-policy enforcement on the sensor streams. Cloudlets thus provide the foundation for a scalable and secure privacy solution that aligns well with natural organizational boundaries of trust and responsibility.

As Figure 3 illustrates, full-fidelity sensor data can be archived on a cloudlet for a finite duration such as a few hours, days, or weeks depending on data volume and local storage size. This could be valuable in case the IoT hub discovers an anomaly and returns a request for more in-depth data analysis using less aggressively denatured data. Whether to relax the normal privacy policy in such situations is a decision that remains under end-user control.

MASKING CLOUD OUTAGES
As our dependence on the cloud grows, so does our vulnerability to cloud outages. Implicit in the convergence of mobile and cloud computing is the assumption that the cloud is easily accessible at all times—in other words, there is good end-to-end network quality and few network or cloud failures. However, there are usage contexts in which cloud access must be viewed as an occasional luxury rather than a basic necessity. This viewpoint applies to several important contexts that can be referred to as *hostile environments*.

The prime example of a hostile environment is a theater of military operations—jamming the enemy's network is a standard tactic. Another example is a geographical region where recovery is under way after networking infrastructure has been destroyed by a natural disaster or terrorist attack. A third example is a developing country with weak networking infrastructure. A fourth example is any part of the Internet that has temporarily become a hostile environment because it is under cyberattack. There is growing concern that cyberattacks could soon become major weapons of organized crime as well as instruments of national policy. If these dire predictions come true, the entire Internet might have to be viewed as a hostile environment in the future.

Cloudlets can alleviate cloud outages. Because of physical proximity, the survivability characteristics of a cloudlet are closer to its associated mobile devices than to the distant cloud. This opens the door to approaches in which a fallback service on a cloudlet can temporarily mask cloud inaccessibility.[17] During failures, a cloudlet can serve as a proxy for the cloud and perform its critical services. Upon repair of the failure, actions that were tentatively committed to the cloudlet might need to be propagated to the cloud for reconciliation.

More than two decades ago, James Kistler and I anticipated how this concept could be applied to cloud-sourced data in describing the Coda File System, which provided disconnectable read–write access to shared data.[18] The essential steps are *hoarding* (prefetching data into a persistent cache), *emulation* (leveraging hoarded data in the cloud's absence and precisely tracking local updates), and *reintegration* (propagating updates to the cloud, and detecting and resolving conflicts). Generalizing these steps to various cloud services will be an important future research area.

[
DURING FAILURES, A CLOUDLET CAN SERVE AS A PROXY FOR THE CLOUD AND PERFORM ITS CRITICAL SERVICES.
]

THE ROAD AHEAD

Edge computing clearly offers many benefits. At the same time, it also faces many technical and nontechnical challenges.

On the technical side, there are many unknowns pertaining to the software mechanisms and algorithms needed for the collective control and sharing of cloudlets in distributed computing. There are also substantial hurdles in managing dispersed cloudlet infrastructure. As mentioned earlier, one of cloud computing's driving forces is the lower management cost of centralized infrastructure. The dispersion inherent in edge computing raises the complexity of management considerably. Developing innovative technical solutions to reduce this complexity is a research priority for edge computing. Another important area of study will be the development of mechanisms to compensate for the weaker perimeter security of cloudlets, relative to cloud datacenters. The development of tamper-resistant and tamper-evident enclosures, remote surveillance, and Trusted Platform Module–based attestation are all important paths that could contribute to alleviating this problem.

On the nontechnical side, the biggest unknown relates to viable business models for deploying cloudlets. Success will require the involvement and support of a complex set of industries, communities, and standards organizations. This presents a classic bootstrapping problem. Without unique applications and services that leverage edge computing, there is no incentive for deploying cloudlets. Yet, without large-enough cloudlet deployments, there is little incentive for developers to create those new applications and services. How can we break this deadlock?

This state of affairs is similar to that at the dawn of the Internet in the late 1970s to early 1980s. An open ecosystem attracted investment in infrastructure and applications, without any single entity bearing large risk or dominating the market. Over time, this lead to the emergence of a critical mass of Internet infrastructure and applications (such as email) that could uniquely benefit from that infrastructure. By the time the World Wide Web emerged as a "killer app" in the early 1990s, sufficient Internet infrastructure had been deployed for growth to explode.

Edge computing can follow a similar, but faster, path to success by nurturing the creation of an open cloudlet ecosystem. This is the goal of

ABOUT THE AUTHOR

MAHADEV SATYANARAYANAN (Satya) is the Carnegie Group Professor of Computer Science at Carnegie Mellon University (CMU). His multidecade research career has focused on the challenges of performance, scalability, availability, and trust in information systems that reach from the cloud to the mobile edge of the Internet. In the course of this work, he pioneered many advances in distributed systems, mobile computing, pervasive computing, the Internet of Things, and, most recently, edge/fog computing. Satya received a PhD in computer science from CMU. He is a Fellow of ACM and IEEE. Contact him at satya@cs.cmu.edu.

OEC's OpenStack++, a derivative of the popular OpenStack cloud computing platform. The "++" refers to the unique extensions necessary for cloudlet environments including cloudlet discovery, just-in-time provisioning, and VM hand-off. As edge computing grows, OpenStack++ aims to become a widely used platform that catalyzes many proprietary and nonproprietary innovations in hardware, software, and services.

The emergence of edge computing coincides with three important trends in the computing and communication landscape that, despite being driven by distinct forces, are convergent. One trend is software-defined networking (SDN) and the associated concept of network function virtualization (NFV), which must be supported by some of the same virtualized infrastructure as edge computing. A second trend is growing interest in ultra-low-latency (one millisecond or less) wireless networks for a new class of tactile applications. Ultra-low latency is one of the proposed attributes for future 5G networks. Edge computing is a natural partner of 5G networks because it ensures that ultra-low first-hop latency is not swamped by the much larger latency of the remaining hops to the cloud. A third trend is continuing improvement in the computing capabilities of wearables, smartphones, and other mobile devices that represent the Internet's extreme edge. Although these devices are indeed growing in computing power, their improvements are muted by the fundamental challenges of mobility such as weight, size, battery life, and heat dissipation. The sweet spot for edge computing is thus in the infrastructure, where it can amplify the capabilities of proximate mobile devices and sensors.

In closing, it is useful to reflect on edge computing from a historical perspective. Since the 1960s, computing has alternated between centralization and decentralization. The centralized approaches of batch processing and timesharing prevailed in the 1960s and 1970s. The 1980s and 1990s saw decentralization through the rise of personal computing. By the mid-2000s, the centralized approach of cloud computing began its ascent to the preeminent position that it holds today. Edge computing represents the latest phase of this ongoing dialectic. ■

REFERENCES

1. M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive*

Computing, vol. 8, no. 4, 2009, pp. 14–23.

2. F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proc. 1st Edition MCC Workshop Mobile Cloud Computing* (MCC 12), 2012, pp. 13–15.

3. "IBM and Nokia Siemens Networks Announce World's First Mobile Edge Computing Platform #MWC13," press release, Nokia, 25 Feb. 2013; company .nokia.com/en/news/press-releases /2013/02/25/ibm-and-nokia-siemens -networks-announce-worlds-first -mobile-edge-computing-platform -mwc13.

4. S. Antipolis, "ETSI Announces First Meeting of New Standardization Group on Mobile-Edge Computing," *The Standard*, European Telecommunications Standards Inst., 30 Oct. 2014; www.etsi.org/news-events /news/838-2014-10-news-etsi -announces-first-meeting-of-new -standardization-group-on-mobile -edge-computing.

5. J. Dilley et al., "Globally Distributed Content Delivery," *IEEE Internet Computing*, vol. 6, no. 5, 2002, pp. 50–58.

6. B.D. Noble et al., "Agile Application-Aware Adaptation for Mobility," *Proc. 16th ACM Symp. Operating Systems Principles* (SOSP 97), 1997, pp. 276–287.

7. J. Flinn and M. Satyanarayanan, "Energy-Aware Adaptation for Mobile Applications," *Proc. 17th ACM Symp. Operating Systems Principles* (SOSP 99), 1999, pp. 48–63.

8. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Comm.*, vol. 8, no. 4, 2001, pp. 10–17.

9. A. Li et al., "CloudCmp: Comparing Public Cloud Providers," *Proc. 10th ACM SIGCOMM Conf. Internet Measurement* (IMC 10), 2010, pp. 1–14.

10. J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *ACM Queue*, vol. 9, no. 11, 2011; queue.acm .org/detail.cfm?id=2071893.

11. K. Ha et al., "Towards Wearable Cognitive Assistance," *Proc. 12th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 14), 2014, pp. 68–81.

12. K. Ha et al., "The Impact of Mobile Multimedia Applications on Data Center Consolidation," *Proc. 2013 IEEE Int'l Conf. Cloud Eng.* (IC2E 13), 2013, pp. 166–176.

13. S. Banerjee and D.O. Wu, *Final Report from the NSF Workshop on Future Directions in Wireless Networking*, Nat'l Science Foundation, Nov. 2013; ecedha.org/docs/nsf-nets/final -report.pdf.

14. Z. Chen et al., "Early Implementation Experience with Wearable Cognitive Assistance Applications," *Proc. 2015 Workshop Wearable Systems and Applications* (WearSys 15), 2015, pp. 33–38.

15. P. Simoens et al., "Scalable Crowd-Sourcing of Video from Mobile Devices," *Proc. 11th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 13), 2013, pp. 139–152.

16. N. Davies et al., "Privacy Mediators: Helping IoT Cross the Chasm," *Proc. 17th Int'l Workshop Mobile Computing Systems and Applications* (HotMobile 16), 2016, pp. 39–44.

17. M. Satyanarayanan et al., "The Role of Cloudlets in Hostile Environments," *IEEE Pervasive Computing*, vol. 12, no. 4, 2013, pp. 40–49.

18. J.J. Kistler and M. Satyanarayanan, "Disconnected Operation in the Coda File System," *ACM Trans. Computer Systems*, vol. 10, no. 1, 1992, pp. 3–25.