

The Seminal Role of Edge-Native Applications

Mahadev Satyanarayanan
Carnegie Mellon University
Pittsburgh, PA, USA
satya@cs.cmu.edu

Guenter Klas, Marco Silva, Simone Mangiante
Vodafone Group
Newbury, UK
{Guenter.Klas, marco.silva1, simone.mangiante}@vodafone.com

Abstract—We introduce the concept of *edge-native applications* that fully exploit the potential of edge computing and have a deeply symbiotic relationship with it. Such an application is custom-designed to take advantage of one or more of the unique attributes of edge computing such as (a) bandwidth scalability, (b) low-latency offload, (c) privacy-preserving denaturing, and (d) WAN-failure resiliency. The application may also contribute to scalability through adaptation to reduce offered load. We contrast edge-native applications with shallower uses of edge computing in a taxonomy that spans *edge-enhanced*, *device-native* applications, *edge-accelerated*, *cloud-native* applications, and *device-only* applications. We close with a case study that illustrates these concepts in the context of cognitive assistance for automotive safety.

I. INTRODUCTION

The roots of Edge Computing reach back over a decade. This new tier of computing has arisen from the observation that *consolidation*, which is the central premise of cloud computing, has negative consequences. It tends to lengthen network round-trip times (RTT) from mobile users, and to increase cumulative ingress bandwidth demand from Internet of Things (IoT) devices. These negative consequences stifle the emergence of new classes of real-time, sensor-rich applications such as assistive augmented reality (AR) and streaming IoT video analytics.

Since we first articulated this insight in 2009 [1], experimental studies by us and others have validated and quantified the many benefits of edge computing. First, we have shown [2], [3], [4], [5] that edge computing can help to achieve considerable improvement in *response times* and *battery life* by offloading computation from a mobile device to a nearby *cloudlet* rather than to the distant cloud. Second, we have demonstrated [6], [7], [8], [9], [10] that enormous reduction in *ingress bandwidth demand* is achievable by processing high data rate sensor streams (such as video streams) on cloudlets rather than the cloud. Third, we have shown [10], [11] that cloudlets can serve as *privacy firewalls* that enable users to dynamically and selectively control the release of sensitive information from sensors to the cloud. Fourth, we have shown [12] that cloudlets can offer *fallback services* that mask the unavailability of cloud services due to network or server failures, or cyber attacks. A decade after its original conception, the importance of edge computing is no longer in doubt. The focus of effort is now on accelerating the adoption of edge computing.

This paper focuses on the transformative aspects of edge computing. We begin in Section II by positioning edge computing relative to other elements of today's computing landscape. Then, in Section III, we identify a broad class of new applications called *edge-native applications* that fully exploit the potential of edge computing and have a deeply symbiotic relationship with it. In Section IV, we develop a taxonomy that contrasts edge-native applications with other applications that make shallower use of edge computing. To illustrate the abstract concepts in our discussion, Section V describes a specific use case and discusses how alternative implementations map to our taxonomy. We close in Section VI by reiterating the central message of this paper: it is edge-native applications that will deliver transformative value to society, and investing in them is critical to business success in edge computing.

II. A THREE-TIER MODEL OF COMPUTING

The cumulative body of evidence cited in Section I on the merits of edge computing has led to the three-tier model shown in Figure 1. Each tier represents a distinct and stable set of design constraints that dominate attention at that tier [13]. There are typically many alternative implementations of hardware and software at each tier, but all of them are subject to the same set of design constraints. There is no expectation of full interoperability across tiers — randomly choosing one component from each tier is unlikely to result in a functional system. Rather, there are many sets of compatible choices across tiers. For example, a single company will ensure that its products at each tier work well with its own products in other tiers, but not necessarily with products of other companies. The tiered model of Figure 1 is thus quite different from the well-known “hourglass” model of interoperability. Rather than defining functional boundaries or APIs, this model segments the end-to-end computing path and highlights design commonalities.

Tier-1 represents “the cloud” in today's parlance. Two dominant themes of Tier-1 are *compute elasticity* and *storage permanence*. Cloud computing has almost unlimited elasticity, as a Tier-1 data center can easily spin up servers to rapidly meet peak demand. Relative to Tier-1, all other tiers have very limited elasticity. In terms of archival preservation, the cloud is the safest place to store data with confidence

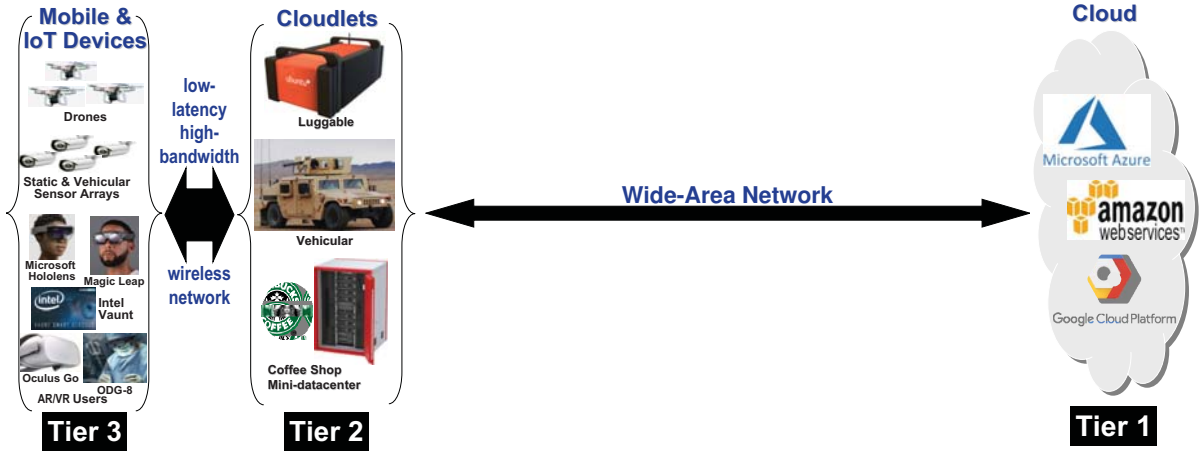


Figure 1. Three-tier Model of Computing

that it can be retrieved far into the future. A combination of storage redundancy (e.g., RAID), infrastructure stability (i.e., data center engineering), and management practices (e.g., data backup and disaster recovery) together ensure the long-term integrity and accessibility of data entrusted to the cloud. Relative to the data permanence of Tier-1, all other tiers offer more tenuous safety. Getting important data captured at those tiers to the cloud is often an imperative. Tier-1 exploits economies of scale to offer very low total costs of computing. As hardware costs shrink relative to personnel costs, it becomes valuable to amortize IT personnel costs over many machines in a large data center. *Consolidation* is thus a third dominant theme of Tier-1. For many large tasks, Tier-1 is typically the optimal place to perform them. This remains true even after the emergence of edge computing.

Mobility and sensing are the defining attributes of Tier-3. Mobility places stringent constraints on weight, size, and heat dissipation of devices that a user carries or wears [14]. Such a device cannot be too large, too heavy or run too hot. Battery life is another crucial design constraint. Together, these constraints severely limit designs. Technological breakthroughs (e.g., a new battery technology) may expand the envelope of feasible designs, but the underlying constraints always remain.

Today’s mobile devices are rich in sensors such as GPS, microphones, accelerometers, gyroscopes, and video cameras. Unfortunately, a mobile device may not be powerful enough to perform real-time analysis of data captured by its on-board sensors (e.g., video analytics). While mobile hardware continues to improve, there is always a large gap between what is feasible on a mobile device and what is feasible on a server of the same technological era. Figure 2 shows this large performance gap persisting over a 20-year period from 1997 to 2017. One can view this stubborn gap as a “mobility penalty” — i.e., the price one pays in performance foregone due to mobility constraints.

To overcome this penalty, a mobile device can offload

| Year | Typical Tier-1 Server | | Typical Tier-3 Device | |
|------|-----------------------|--------------------------|-----------------------|-----------------------|
| | Processor | Speed | Device | Speed |
| 1997 | Pentium II | 266 MHz | Palm Pilot | 16 MHz |
| 2002 | Itanium | 1 GHz | Blackberry 5810 | 133 MHz |
| 2007 | Intel Core 2 | 9.6 GHz (4 cores) | Apple iPhone | 412 MHz |
| 2011 | Intel Xeon X5 | 32 GHz (2x6 cores) | Samsung Galaxy S2 | 2.4 GHz (2 cores) |
| 2013 | Intel Xeon E5-2697v2 | 64 GHz (2x12 cores) | Samsung Galaxy S4 | 6.4 GHz (4 cores) |
| | | | Google Glass | 2.4 GHz (2 cores) |
| 2016 | Intel Xeon E5-2698v4 | 88.0 GHz (2x20 cores) | Samsung Galaxy S7 | 7.5 GHz (4 cores) |
| | | | HoloLens | 4.16 GHz (4 cores) |
| 2017 | Intel Xeon Gold 6148 | 96.0 GHz (2x20 cores) | Pixel 2 | 9.4 GHz (4 cores) |

Source: Adapted from Chen [15] and Flinn [16]
“Speed” metric = number of cores times per-core clock speed.

Figure 2. The Mobility Penalty: Impact of Tier-3 Constraints

computation over a wireless network to Tier-1. This was first described by Noble et al [17] in 1997, and has since been extensively explored by many others [16], [18]. For example, speech recognition and natural language processing in iOS and Android nowadays work by offloading their compute-intensive aspects to the cloud.

IoT devices can be viewed as Tier-3 devices. Although they may not be mobile, there is a strong incentive for them to be inexpensive. Since this typically implies meager processing capability, offloading computation to Tier-1 is again attractive.

The introduction of Tier-2 is the essence of *edge computing* [19], and creates the illusion of bringing Tier 1 “closer.” This achieves two things. First, it enables Tier 3 devices to offload compute-intensive operations at very low latency. This helps to overcome stringent Tier 3 design constraints (e.g., weight, size, battery life, heat dissipation) without

compromising the tight response time bounds needed for immersive user experience and by cyber-physical systems. Proximity also results in a much smaller fan-in between Tiers 3 and 2 than was the case when Tier 3 devices connected directly to Tier 1. Consequently, Tier 2 processing of large volumes of live data captured at Tier 3 avoids excessive bandwidth demand anywhere in the system.

Note that “proximity” here refers to *network proximity* rather than physical proximity. It is crucial that RTT be low and end-to-end bandwidth be high. This is achievable by using a fiber link between a wireless access point and a cloudlet that is many tens or even hundreds of kilometers away. Conversely, physical proximity does not guarantee network proximity. A highly congested WiFi network may have poor RTT, even if Tier-2 is physically near Tier-3.

III. EDGE-NATIVE APPLICATIONS

Edge computing offers at least four valuable attributes, as mentioned in Section I: (a) bandwidth scalability, (b) low-latency offload, (c) privacy-preserving denaturing, and (d) WAN-failure resiliency. An *edge-native application* is one that is custom-designed to take advantage of one or more of these attributes. Such an application does not function satisfactorily without a cloudlet. It is from this class of applications, uniquely enabled by the attributes above, that the “killer apps” of edge computing are going to emerge.

Even an imperfect initial implementation of a future killer app can provide such high value to end users that it creates new demand for edge computing. In the absence of viable competing alternatives, such an application can coevolve with supporting infrastructure over an extended period of time. For example, there is strong evidence that the development of the spreadsheet circa 1982-1983 (VisiCalc, Lotus-123 and, eventually, Microsoft Excel) was a major catalyst in the adoption of personal computers (PCs) by small businesses. It was the low and stable latency of human interaction (relative to timesharing) that made PCs indispensable infrastructure for spreadsheets.

The four attributes listed above have the potential to play an analogous role for edge computing. The creation of new edge-native applications that leverage one or more of these attributes will be the true drivers of edge computing. The history of science and technology is full of examples of rudimentary implementations of “killer apps” (e.g., automobiles, aircraft, television, the microprocessor, the World Wide Web) driving the evolution of the surrounding ecosystem. This rapidly establishes a virtuous cycle that leads to continuous long-term improvements and business value in both the core technology and the sustaining ecosystem.

Edge-native applications that *augment human cognition* [2], [3], [20], [21] are potential killer apps for edge computing. These applications improve some aspect of human cognition (e.g., task performance, long-term memory, face recognition, etc.) in real time. By leveraging edge

computing, the computing resources that can be brought to bear in this task can be far larger, heavier, more energy-hungry and more heat-dissipative than could ever be carried or worn by a human user. Distributed sensing can also offer real-time sensory inputs (e.g., from video cameras) obtained from vantage points other than the first-person viewpoint of a human. By seamlessly integrating these resources with human perception and cognition, such an assistive application could achieve a whole that is much greater than the sum of parts.

The potential business value of cognitive augmentation is huge. Consider, for example, just one small segment of this market: cognitive assistance for the elderly in the US. Today, over 20 million Americans are affected by some form of cognitive decline that significantly affects their ability to function as independent members of society. This includes people with neurodegenerative conditions such as Alzheimer’s disease (~4.5M) and mild cognitive impairment (>6M), survivors of stroke (~2.5M) and people with traumatic brain injury (~5.3M). These numbers are expected to grow significantly due to an aging population and an increase in long-term post-traumatic stress disorders arising from occupational and social causes. Cognitive impairment can manifest itself in many ways, including the inability to recognize people, locations and objects, loss of short- and long-term memory, and changes in behavior and appearance such as decreased attention to personal hygiene. Among the many challenges faced by older Americans, cognitive decline often has the largest negative impact on them and their family members. The potential cost savings from even modest steps towards addressing this challenge are enormous. In the US alone, it is estimated that just a one-month delay in nursing home admissions nationwide could save over \$1 billion annually [22]! At global scale, with rapidly aging populations in the richest countries of the world, the potential business value is many times larger.

Of course, cognitive augmentation is not the only source of edge-native applications. Real-time video analytics from a large array of cameras is another example, with many uses in domains such as law enforcement, surveillance, and military intelligence. Combining live video analytics with real-time denaturing for privacy is an even more demanding source of edge-native applications [10]. 360-degree video/VR distribution, as reported by Mangiante et al [23] is another example of an edge-native application. Over time, we are confident that the many valuable attributes of edge computing will lead to many different types of edge-native applications. We emphasize cognitive augmentation in this paper because it has been a major source of first-hand experience for us, and because we believe that such applications will ultimately have the potential for major societal benefit.

Our characterization of edge-native applications until this point has focused solely on their deep dependence on edge-specific attributes that cannot be offered by cloud computing.

However, a complete definition of edge-native applications would also have to recognize the fact that they need to be written to function effectively in spite of limitations of the edge. In particular, *scalability* is an area where edge computing suffers relative to cloud computing. As its name implies, a cloudlet is designed for much smaller physical space and electrical power than a cloud data center. Hence, the sudden arrival of an unexpected flash crowd of users can overwhelm a cloudlet and its wireless network. There are only two solutions to this problem. The first is to relocate one or more application back-ends to a less-loaded cloudlet using a mechanism such as VM Handoff [24]. Unfortunately, this is likely to be suboptimal if the original cloudlet was optimally chosen. The other solution is for applications associated with that cloudlet to lower their resource demands. In other words, scalability requires the average burden placed by each user on the cloudlet and the wireless channel to fall as these resources saturate.

These considerations suggest that *adaptive application behavior* based on guidance received from the cloudlet, the network, or inferred by the user's mobile device needs to be an integral part of what it means to be an edge-native application. This important area of future research in edge computing can build upon earlier work. Specifically, the approach to trading off quality of user experience or battery life for reduced resource demand that was introduced by Odyssey [17], [25], and the concept of *multi-fidelity applications* [26], are both valuable foundations to build upon. Scalability at the edge is thus only achievable for applications that have been designed with this goal in mind. This is another important aspect of edge-native applications.

IV. A TAXONOMY OF EDGE COMPUTING APPLICATIONS

There is intense industry interest in edge computing, and it is believed that we are on the cusp of major industry investments [27]. Coming at the same time as the rollout of 5G wireless systems, the total demand for capital investments is substantial even for large telecommunications companies. A question that is frequently asked is whether 5G should be a precursor to edge computing. For a number of use cases, deploying 5G without also deploying edge computing is unlikely to be satisfactory. In such a deployment, 5G only improves last-mile connectivity. The rest of the path to the distant cloud remains what it is today. Average RTTs on the order of a hundred milliseconds, with a tail of one second or more, can be expected [28]. This is a lower bound on the achievable end-to-end application response time. In contrast, deploying edge computing in today's 4G LTE environment can yield end-to-end application response times (both mean and tail) of just a few tens of milliseconds [4], [5]. Edge computing cuts RTT for both 4G LTE and 5G, and makes innovative applications already viable on 4G networks today.


In spite of enormous industry interest in edge computing, actual deployments of edge computing are minimal today

(May 2019) and the path forward is shrouded in uncertainty. The source of this uncertainty is a "chicken or egg" problem involving customer demand for edge computing. Before making large investments, companies expect to see clear evidence of customer demand for edge computing. This demand will only arise when there are a large number of edge-native applications that deliver high value to end users. Before investing resources to create such applications, their authors need confidence that the critical resource (i.e., edge computing) is widely deployed. This is the deadlock we face.



There are two paths to breaking this deadlock, and we assert that both paths are important. First, the industry segments that stand to benefit from edge computing over the long term should play an active role in nurturing the creation of edge-native applications. This could be through support of non-commercial open source efforts, as well as direct investments in startup companies. In making these investments, the potential beneficiaries of a vibrant edge computing ecosystem should recognize that "return on investment (ROI)" should be interpreted more broadly than classic investment metrics would suggest. They should take the long view, and recognize that ROI includes their own long-term survival. This requires an active role in application development, and lies well outside the traditional comfort zone of many companies. Especially for telecommunications companies, edge computing represents an opportunity to create long term business value for their unique assets.

The second path is to recognize that edge-native applications are not the only class of applications to benefit from edge computing. There are applications that can leverage edge computing when available, but deliver acceptable user experience when run in the cloud. We refer to this class of applications as *edge-accelerated, cloud-native applications*. For brevity, we just use the qualifier "edge-accelerated" to refer to this class. The 20-year history of content distribution networks (CDNs) for web access is a good example of edge acceleration. Today, venture capital is focused on identifying new edge-accelerated use cases rather than edge-native use cases, because they involve less investment risk. Edge-accelerated use cases involve much less software development, and their markets are much larger since they can function acceptably even in the absence of edge computing. To the extent that they stimulate investment, these applications can offer modest help in catalyzing the rollout of edge infrastructure. However, unlike edge-native applications, they will not deliver transformative value and will therefore generate only modest premium for the edge.

Between edge-native and edge-accelerated applications are a third class of applications that leverage edge computing. These typically run on-device, rather than in the cloud. When edge computing is available, they are able to make optional use of it to improve functionality or performance. For example, an application may have certain features that only work when the device is associated with a cloudlet.



| | Tier-3 | Tier-2 | Tier-1 |
|--------------------------------|--------|--------|--------|
| Device-only | P | | |
| Edge-accelerated, cloud-native | | O | P |
| Edge-enhanced, device-native | P | O | |
| Edge-native | P | P | |

 Primary execution site
 Optional, non-critical use

This table only refers to the main code paths. Not shown are incidental uses of Tier-1 for purposes such as authentication, error reports, and software upgrades.

Figure 3. Taxonomy of Application Types

Otherwise, only a subset of the full functionality may be available on the device. Another approach is to reduce the fidelity of the underlying algorithms when edge computing is not available. This approach is consistent with the approach of *multi-fidelity computation*, introduced nearly 20 years ago [26]. In a computer vision application, for example, the algorithm used for object detection may be much more accurate on the cloudlet. The version of the application on the mobile device may use a simpler algorithm that has smaller memory footprint and lower processing demand, but comes at the cost of lower accuracy. We refer to this class of applications as *edge-enhanced, device-native* applications. For brevity, we just use the qualifier “edge-enhanced” to refer to this class.

Finally, there are many applications that run entirely on a Tier-3 device, without any use of Tier-2. We refer to these as *device-only* applications. The opportunity here is to evangelize edge computing, and to help developers to refine and extend these applications into edge-native or edge-enhanced, device-native applications that offer improved functionality, user-experience or both.

Figure 3 visually illustrates these four classes of applications. Edge computing is more deeply used as we go from top to bottom. At the very bottom (edge-native), one or more attributes of edge computing is so deeply woven into the fabric of the application that it cannot function effectively without edge computing. As we move up through the edge-enhanced and edge-accelerated levels of this table, the dependence on edge computing decreases. It is deep dependence on edge computing that will ultimately drive its widespread deployment and success. Relative to scalability, only edge-native applications contribute to reducing offered load through adaptation. They can thus be viewed as the true first-class citizens of the edge, both contributing to and benefiting from its unique attributes.

V. CASE STUDY

A. Concept

To illustrate the concepts discussed in Sections III and IV, we present a real-world use case. This is in the domain of *automotive safety*. Specifically, it is a cognitive assistance application for drivers (and future autonomous vehicles) that alerts them to a pedestrian who unexpectedly appears in front of a moving vehicle. In theory, a driver should always be on the alert for such a possibility. In practice, many factors such as fatigue, poor lighting, and distracted driving (e.g., talking or texting on a cell phone) together conspire to make drivers less than perfect in their ability to avoid accidents in this setting. The goal is to create a cognitive assistant that uses computer vision and edge computing to detect the pedestrian and immediately alert the driver. The input for sensing comes from a video camera mounted on the vehicle. Such a system would, in effect, augment the driver’s vision system with an extra pair of eyes that never gets fatigued or distracted. Figure 4 illustrates this concept.

Our use of edge computing has at least two benefits. First, it avoids the need for expensive in-vehicle compute infrastructure to process frames from the video camera in real time. For a low-end vehicle costing \$20,000, even a \$1,000 computer is a significant price increase. The drawback in having no in-vehicle cloudlet is the significant continuous use of wireless bandwidth to stream video to an off-vehicle cloudlet. In recent work [29], we have shown how preliminary on-board processing of the video by a relatively small and cheap cloudlet can suppress transmission of many video frames without loss of accuracy in detection. Such a hybrid approach may strike the right cost balance.

Second, looking to the future, it may be possible to combine video and GPS data from a vehicle with additional video streams from static cameras nearby to improve the predictive ability of the system. For example, a camera pointed at the sidewalk may detect a soccer ball kicked towards the road. AI software can reasonably infer that the child who kicked the ball may run onto the street to retrieve it. It can therefore proactively raise an alert before the child actually runs onto to the road. The window of advance notice may only be a few hundred milliseconds, but that may save the child’s life. A cloudlet that is processing all these video streams views a vehicle’s video stream as just one more input in its sensor fusion. It will be difficult in a purely vehicle-centric system to leverage these multiple sensor viewpoints in the surrounding infrastructure.

Others have also recognized the value of edge computing for real-time video analytics in traffic safety. For example, Ananthanarayanan et al [30] have described a video analytics software stack for edge computing that has been deployed in the city of Bellevue, WA and aims to provide high accuracy while minimizing the cost of execution. In our taxonomy, their system would be viewed as an edge-native application.

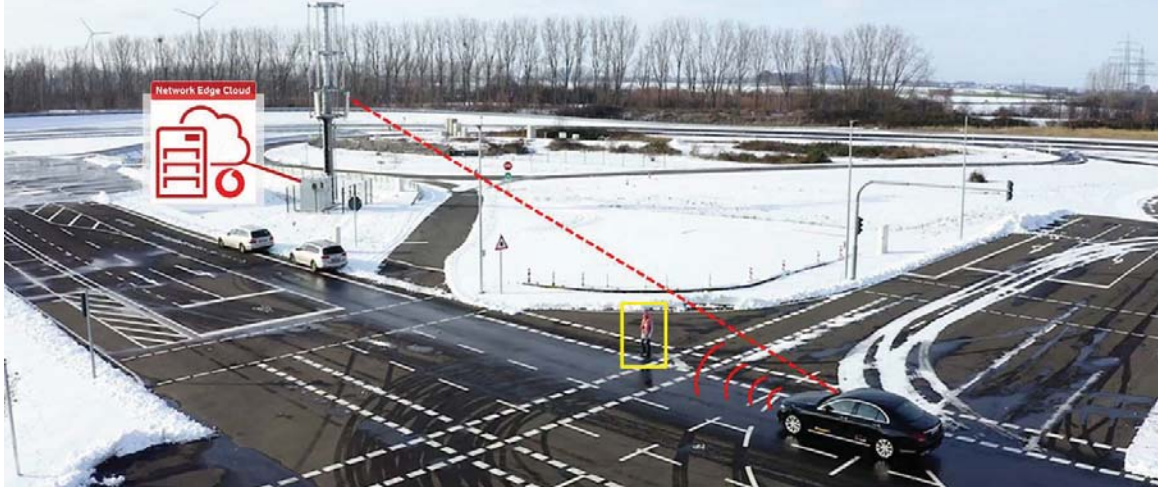


Figure 4. Predictive Automotive Safety through AI at the Edge



Figure 5. Aldenhoven Testing Center, Germany



Figure 6. One of the Cloudlets Used in the Prototype System

B. Proof of Concept Implementation

In partnership, Vodafone (a leading telecommunications company) and Continental (a leading technology company) have created a proof of concept implementation of this system. The implementation is located at the outdoor automotive test track of the Aldenhoven Testing Center in Germany (Figure 5), the home of Vodafone’s 5G Mobility lab. Continental developed the application software; Vodafone and its vendors created the edge computing infrastructure (Figure 5, 6). This infrastructure features a MEC (“multi-access edge

computing”) software platform, Software Defined Networking Control, virtualised IPSec gateways, virtualized active measurement software and probes, and a Vodafone-specific cloudlet environment supporting cloud native development and hardware acceleration through GPUs. A series of drive tests were conducted to analyse robustness of the application in the face of varying radio conditions, radio cell traffic load and vehicle speeds. These tests confirmed the viability of this system with 4G LTE. Upgrading to 5G will boost available bandwidth per vehicle, while also reducing radio latency.

C. Mapping to Taxonomy

As implemented in February 2019, this system is very much dependent on the cloudlet — only a little pre-processing is done in the vehicle. The low end-to-end latency offered by edge computing is crucial in this application. At typical vehicular speeds, even a few hundred milliseconds can mean the difference between an accident and a near miss: for example, at 30 mph, a vehicle covers 4.4 feet in 100 milliseconds.

Edge computing is also crucial for bandwidth scalability in this implementation. Hulu estimates that its video streams require 13 Mbps for 4K resolution and 6 Mbps for HD resolution using highly optimized offline encoding [31]. Live streaming is less bandwidth-efficient: 10 Mbps for HD video at 25 FPS is reported by Wang et al [32]. Without edge computing, the total bandwidth demand to the cloud from 10,000 vehicles in a city would exceed 100 Gbps. This would place severe stress on its metropolitan area network.

In the taxonomy of Section IV, this version of the application qualifies as an edge-native application since it is critically dependent on one or more attributes of edge computing. However, it lacks any support for scalability at the edge. If a large number of proximate vehicles overwhelm the processing capacity of the single cloudlet that they are associated with, there is currently no mechanism to

reduce offered load. This would require one of three possible solutions to be added. First, some vehicles could be denied cognitive assistance (load shedding). Second, the prediction algorithm used in the cloudlet could be temporarily switched to a less accurate (and presumably less compute intensive) algorithm. Third, an in-vehicle cloudlet could process many of the frames with some loss of accuracy and only transmit a much lower frame rate to the cloudlet. The first and third approaches could also be used to reduce wireless bandwidth demand, if that proves to be the bottleneck.

How would alternative implementations of this application map to the taxonomy of Figure 3? Based on the latency and bandwidth scalability reasoning above, it is clear that this application simply cannot be run in the cloud. There is therefore no flavor of implementation that could be considered as an edge-accelerated, cloud-native implementation.

A device-only implementation is conceivable. A large in-vehicle cloudlet with a high-end GPU could provide the computing cycles necessary to perform video processing at full frame rate. This would offer the lowest possible latency, since no wireless communication to the cloudlet would be required. Such a design would also be highly bandwidth scalable. The biggest drawback of a device-only implementation is the significant cost of an in-vehicle cloudlet that is powerful enough for continuous video analytics at high frame rate. A secondary drawback is the inability to leverage video streams from static cameras in the surroundings, and to thus benefit from sensor fusion.

A slight modification of the device-only implementation, to allow use of a cloudlet when available, could eliminate this drawback. Such an implementation would rely solely on the in-vehicle camera and cloudlet in the worst case, but leverage cloudlet-based sensor fusion when possible. This would be viewed as an edge-enhanced implementation in our taxonomy. However, if this implementation were to also be adaptive and to reduce offered load when the wireless network is congested or the cloudlet is heavily loaded, it would be more appropriate to view it as a full-fledged edge-native application.

VI. CONCLUSION

A decade ago, the emergence of cloud computing led to the concept of cloud-native applications. These were applications that were designed and implemented from the ground up to take full advantage of a unique attribute of the cloud, namely its extreme elasticity. A cloud data center has almost unlimited capacity to spin up more servers to meet peak demand. However, to take advantage of this capability, applications have to be written in a particular style which is the essence of a cloud-native application.

By analogy, edge-native applications are written to conform to the unique strengths and weaknesses of edge computing. The strengths include low latency, bandwidth scalability, enhanced privacy, and improved resiliency to WAN

network failures. The main weakness of edge computing is the limited elasticity of even large cloudlets relative to hyperscale cloud data centers. A secondary weakness is the increased cost of managing dispersed rather than centralized infrastructure. This secondary weakness is not directly visible to applications, but manifests itself in the high marginal cost of edge computing relative to cloud computing. This leads to the need for an “edge premium.”

The central message of this paper is that edge-native applications and edge computing need each other. Deploying edge computing without a substantial body of edge-native applications is unlikely to result in a sustainable business. The capital cost of deploying edge infrastructure, and the higher marginal operating cost of that infrastructure relative to a cloud data center, can only be recouped with a substantial premium for edge computing. Such a premium is unlikely to be sustainable unless end-users receive new value that delights them. Only edge-native applications can provide that value. The taxonomy that we have introduced includes other types of applications that can benefit from the edge, without being deeply dependent upon it. While those may be useful fellow travellers in the journey to deployment of edge computing everywhere, we posit that only edge-native applications can help us complete that journey.

ACKNOWLEDGMENTS

Satyanarayanan was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0051 and by the National Science Foundation (NSF) under grant number CNS-1518865. He received additional support from Intel, Vodafone, Deutsche Telekom, Verizon, Crown Castle, NTT, Seagate, Siemens, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view(s) of their employers or the above-mentioned funding sources.

REFERENCES

- [1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, 2009.
- [2] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, P. Pillai, R. Klatzky, D. Siewiorek, and M. Satyanarayanan, “An Empirical Study of Latency in an Emerging Class of Edge Computing Applications for Wearable Cognitive Assistance,” in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, Fremont, CA, October 2017.
- [3] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, “Towards Wearable Cognitive Assistance,” in *Proceedings of ACM MobiSys*, Bretton Woods, NH, June 2014.
- [4] K. Ha, P. Pillai, G. Lewis, S. Simanta, S. Clinch, N. Davies, and M. Satyanarayanan, “The Impact of Mobile Multimedia Applications on Data Center Consolidation,” in *Proceedings of the IEEE International Conference on Cloud Engineering*, San Francisco, CA, 2013.

- [5] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan, "Quantifying the Impact of Edge Computing on Mobile Applications," in *Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*, Hong Kong, China, 2016.
- [6] W. Hu, Z. Feng, Z. Chen, J. Harkes, P. Pillai, and M. Satyanarayanan, "Live Synthesis of Vehicle-Sourced Data Over 4G LTE," in *Proceedings of the 20th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Miami Beach, FL, November 2017.
- [7] M. Satyanarayanan, "Edge Computing for Situational Awareness," in *Proceedings of the 23rd IEEE Symposium on Local and Metropolitan Area Networks*, Osaka, Japan, June 2017.
- [8] M. Satyanarayanan, P. Gibbons, L. Mummert, P. Pillai, P. Simoens, and R. Sukthankar, "Cloudlet-based Just-in-Time Indexing of IoT Video," in *Proceedings of the IEEE 2017 Global IoT Summit*, Geneva, Switzerland, June 2017.
- [9] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable Crowd-Sourcing of Video from Mobile Devices," in *Proc. of ACM MobiSys*, Taipei, Taiwan, 2013.
- [10] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "A Scalable and Privacy-Aware IoT Service for Live Video Analytics," in *Proceedings of ACM Multimedia Systems*, Taipei, Taiwan, June 2017.
- [11] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, and B. Amos, "Privacy Mediators: Helping IoT Cross the Chasm," in *Proc. of ACM HotMobile 2016*, St. Augustine, FL, February 2016.
- [12] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The Role of Cloudlets in Hostile Environments," *IEEE Pervasive Computing*, vol. 12, no. 4, October-December 2013.
- [13] M. Satyanarayanan, W. Gao, and B. Lucia, "The Computing Landscape of the 21st Century," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, February 2019.
- [14] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, 1996.
- [15] Z. Chen, "An Application Platform for Wearable Cognitive Assistance," Ph.D. dissertation, Computer Science Dept., Carnegie Mellon Univ., 2018.
- [16] J. Flinn, *Cyber Foraging: Bridging Mobile and Cloud Computing via Opportunistic Offload*. Morgan & Claypool, 2012.
- [17] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker, "Agile Application-Aware Adaptation for Mobility," in *Proc. of the 16th ACM Symp. on Operating Systems Principles*, 1997.
- [18] M. Satyanarayanan, "A Brief History of Cloud Offload," *ACM GetMobile*, vol. 18, no. 4, 2014.
- [19] —, "The Emergence of Edge Computing," *IEEE Computer*, vol. 50, no. 1, 2017.
- [20] —, "Augmenting Cognition," *IEEE Pervasive Computing*, vol. 3, no. 2, April-June 2004.
- [21] M. Satyanarayanan and N. Davies, "Augmenting Cognition through Edge Computing," *IEEE Computer*, vol. 52, no. 7, July 2019.
- [22] T. Kanade, "Quality of Life Technology," *IEEE Proceedings*, vol. 100, no. 8, August 2012.
- [23] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "VR is on the edge: How to deliver 360 videos in mobile networks," Los Angeles, CA, August 2017.
- [24] K. Ha, Y. Abe, T. Eiszler, Z. Chen, W. Hu, B. Amos, R. Upadhyaya, P. Pillai, and M. Satyanarayanan, "You Can Teach Elephants to Dance: Agile VM Handoff for Edge Computing," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, Fremont, CA, 2017.
- [25] J. Flinn and M. Satyanarayanan, "Energy-aware Adaptation for Mobile Applications," in *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*, Charleston, SC, 1999.
- [26] M. Satyanarayanan and D. Narayanan, "Multi-Fidelity Algorithms for Interactive Mobile Applications," in *Proceedings of the 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM)*, Seattle, WA, August 1999.
- [27] Editorial, "Take it to the edge," *Nature Electronics*, vol. 2, no. 1, January 2019.
- [28] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: comparing public cloud providers," in *Proceedings of the 10th annual conference on Internet measurement*, 2010.
- [29] K. Christensen, C. Mertz, P. Pillai, M. Hebert, and M. Satyanarayanan, "Towards a Distraction-free Waze," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, February 2019.
- [30] G. Ananthanarayanan, P. Bahl, P. Bodik, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-Time Video Analytics: The Killer App for Edge Computing," *IEEE Computer*, vol. 50, no. 10, October 2017.
- [31] Hulu, "Internet speed requirements for streaming HD and 4K Ultra HD," <https://help.hulu.com/en-us/requirements-for-hd>, 2017, Last accessed: May 16, 2017.
- [32] J. Wang, Z. Feng, Z. Chen, S. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, "Bandwidth-efficient Live Video Analytics for Drones via Edge Computing," in *Proc. of the Third IEEE/ACM Symposium on Edge Computing*, Bellevue, WA, October 2018.