Scaling Wearable Cognitive Assistance

Junjue Wang
junjuew@cs.cmu.edu

June 2018

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Mahadev Satyanarayanan (Satya) (Chair)
Daniel Siewiorek
Martial Hebert
Roberta Klatzky
Padmanabhan Pillai (Intel Labs)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Contents

Chapter 1

Introduction

It has been a long endeavour to augment human cognition with machine intelligence. As early as in 1945, Vannevar Bush envisioned a machine Memex that provides "enlarged intimate supplement to one's memory" and can be "consulted with exceeding speed and flexibility" in the seminal article *As We May Think* [1]. This vision has been brought closer to reality by years of research in computing hardware, artificial intelligence, and human-computer interaction. In late 90s to early 2000s, Smailagic et al. [22] [23] [21] created prototypes of wearable computers to assist several cognitive tasks, for example, displaying inspection manuals in a head-up screen to facilitate aircraft maintenance. Around the same time, Loomis et al. [11] [10] explored using computers carried in a backpack to help the blind navigate through auditory cues. Davis et al. [5] [4] developed a context-sensitive intelligent visitor guide leveraging hand-portable multimedia systems. While these research work pioneered cognitive assistance and its related fields, their robustness and functionality are limited by the technologies of their time.

More recently, as underlying technologies experience significant advancement, a new genre of applications, Wearable Cognitive Assistance (WCA) [8] [2], has emerged that pushes the boundaries of augmented cognition. WCA applications continuously process data from body-worn sensors and provide just-in-time guidance to help a user complete a specific task. For example, an IKEA Lamp assistant [2] has been built to assist the assembly of a table lamp. To use the application, a user wears a head-mounted smart glass that continuously captures her actions and surroundings from a first-person viewpoint. In real-time, the camera stream is analyzed to identify the state of the assembly. Audiovisual instructions are generated based on the detected state. The instructions either demonstrate a subsequent procedure or alert and correct a mistake.

Since its conceptualization in 2004 [15], WCA has attracted many research interests from both academia and industry. The building blocks for its vision came into place by 2014, enabling the first implementation of this concept in *Gabriel* [8]. In 2017, Chen et al [3] described a number of applications of this genre, quantified their latency requirements, and profiled the end-to-end latencies of their implementations. In late 2017, SEMATECH and DARPA jointly funded $27.5 million of research on such applications [13, 24]. At the Mobile World Congress in February 2018, wearable cognitive assistance was the focus of an entire session [14]. For AI-based military use cases, this class of applications is the centerpiece of "Battlefield 2.0" [6]. By mid-2019, WCA

was being viewed as a prime source of "killer apps" for edge computing [16, 20].

Different from previous research efforts, the design goals of WCA advance the frontier of mobile computing in multiple aspects. First, wearable devices, particularly head-mounted smart glasses, are used to reduce the discomfort caused by carrying a bulky computation device. Users are freed from holding a smartphone and therefore able to interact with the physical world using both hands. The convenience of this interaction model comes at the cost of constrained computation resources. The small form-factor of smart glasses significantly limits their onboard computation capability due to size, cooling, and battery life reasons. Second, placed at the center of computation is the unstructured high-dimensional image and video data. Only these data types can satisfy the need to extract rich semantic information to identify the progress and mistakes a user makes. Furthermore, state-of-art computer vision algorithms used to analyze image data are both compute-intensive and challenging to develop. Third, many cognitive assistants give real-time feedback to users and have stringent end-to-end latency requirements. An instruction that arrives too late often provides no value and may even confuse or annoy users. This latency-sensitivity further increases their high demands of system resource and optimizations.

To meet the latency and the compute requirements, previous research leverages edge computing and offloads computation to a cloudlet. A cloudlet [18] is a small data-center located at the edge of the Internet, one wireless hop away from users. Researchers have developed an application framework for wearable cognitive assistance, named Gabriel, that leverages cloudlets, optimizes for end-to-end latency, and eases application development [2] [8] [3]. On top of Gabriel, several prototype applications have been built, such as Ping-Pong Assistance, Lego Assistance, Sandwich Assistance, and Ikea Lamp Assembly Assistance. Using these applications as benchmarks, Chen et al. [3] presented empirical measurements detailing the latency contributions of individual system components. Furthermore, a multi-algorithm approach was proposed to reduce the latency of computer vision computation by executing multiple algorithms in parallel and conditionally selecting a fast and accurate algorithm for the near future.

While previous research has demonstrated the technical feasibility of wearable cognitive assistants and meeting latency requirements, many practical concerns have not been addressed. First, previous work operates the wireless networks and cloudlets at low utilization in order to meet application latency. The economics of practical deployment preclude operation at such low utilization. In contrast, resources are often highly utilized and congested when serving many users. How to efficiently scale Gabriel applications to a large number of users remains to be answered. Second, previous work on the Gabriel framework reduces application development efforts by managing client-server communication, network flow control, and cognitive engine discovery. However, the framework does not address the most time-consuming parts of creating a wearable cognitive assistance application. Experience has shown that developing computer vision modules that analyze video feeds is a time-consuming and painstaking process that requires special expertise and involves rounds of trial and error. Developer tools that alleviate the time and the expertise needed can greatly facilitate the creation of these applications.

1.1   Thesis Statement

In this thesis, we address the problem of scaling wearable cognitive assistance. Scalability here has a two-fold meaning. First, a scalable system should enable the most associated clients with fixed amount of infrastructure and has ways to serve more clients as resources increase. Second, we want to enable small software team to quickly create, deploy, and manage these applications. Notably, we claim that:

Two critical challenges to the widespread adoption of wearable cognitive assistance are 1) the need to operate cloudlets and wireless network at low utilization to achieve acceptable end-to-end latency 2) the level of specialized skills and the long development time needed to create new applications. These challenges can be effectively addressed through system optimizations, functional extensions, and the addition of new software development tools to the Gabriel platform.

The main contributions of this thesis are as follows:

1. We propose application-agnostic and application-aware techniques to reduce offered load from WCA mobile client when oversubscription occurs.

2. We provide a profiling-based cloudlet resource allocation mechanism that takes into account of applications' adapatation behaviors.

3. We create a suite of development and deployment tools to reduce the time and lower the barrier of entry for WCA developments.

1.2   Thesis Overview

The remainder of this thesis is organized as follows.

- In Chapter **??**, we describe

Chapter 2

Background

2.1 Edge Computing

Edge computing is a nascent computing paradigm that has gained considerable traction over the past few years. It champions the idea of placing substantial compute and storage resources at the edge of the Internet, in close proximity to mobile devices or sensors. Terms such as "cloudlets" [17], "micro data centers (MDCs)" [?], "fog" [?], and "mobile edge computing (MEC)" [?] are used to refer to these small, edge-located computing nodes. We use these terms interchangably in the rest of this thesis. Edge computing is motivated by its potential to improve latency, bandwidth, and scalability over a cloud-only model. More practically, some efforts stem from the drive towards software-defined networking (SDN) and network function virtualization (NFV), and the fact that the same hardware can provide SDN, NFV, and edge computing services. This suggests that infrastructure providing edge computing services may soon become ubiquitous, and may be deployed at greater densities than content delivery network (CDN) nodes today.

Satya et al. [19] best describes the modern computing landscape with edge computing using a tiered model, shown in Figure 2.1. Tiers are separated by distinct yet stable sets of design constraints. From left to right, this tiered model represents a hierarchy of increasing physical size, compute power, energy usage, and elasticity. Tier-1 represents today's large-scale and heavily consolidated data-centers. Compute elasticity and storage permanence are two dominating themes here. Tier-3 represents IoT and mobile devices, which are constrained by their physical size, weight, and heat dissipation. Sensing is the key functionality of tier-3 devices. For example, today's smartphones are already rich in sensors, including camera, microphone, accelerometers, gyroscopes and GPS. In addition, an increasing amount of IoT devices with specific sensing modalities are getting adopted, e.g. smart speakers, security cameras, and smart thermostats.

With the large-scale deployment of tier-3 devices, there exists a tension between the gigantic amount of data collected and generated by them and their limited capabilities to process these data on-board. For example, most surveillance cameras are limited in computation to run state-of-art computer vision algorithms to analyze the videos they capture. To overcome this tension, a
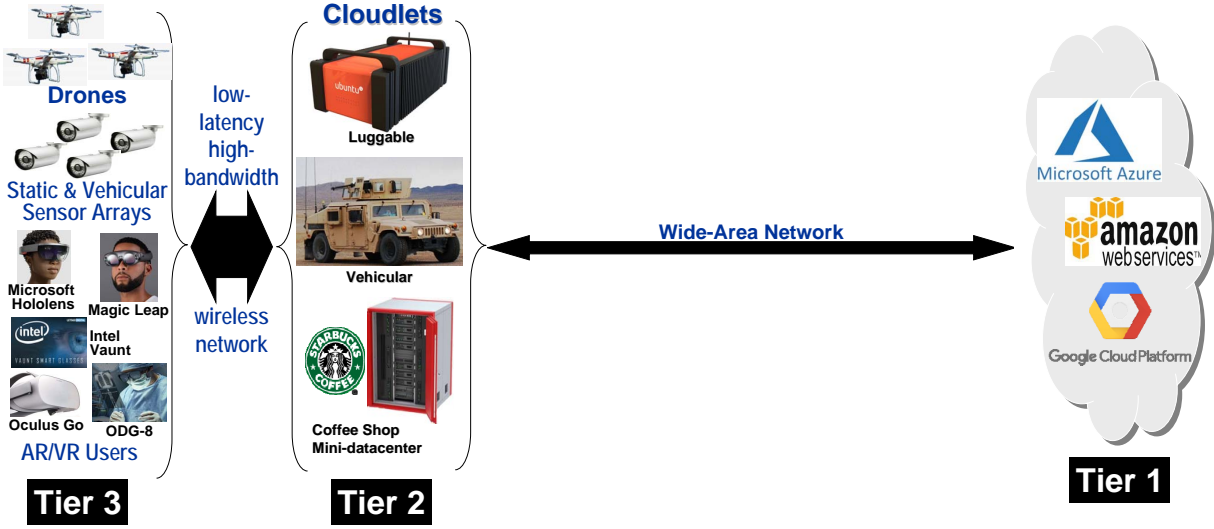
Figure 2.1: Tiered Model of Computing

tier-3 device could offload computation over network to tier-1. This capability was first demonstrated in 1997 by Noble et al. [12], who used it to implement speech recognition with acceptable performance on a resource-limited mobile device. In 1999, Flinn et al. [7] extended this approach to improve battery life. These concepts were generalized in a 2001 paper that introduced the term *cyber foraging* for the amplification of a mobile device's data or compute capabilities by leveraging nearby infrastructure [?]. Thanks to these research efforts, computation offloading is widely used by IoT devices today. For example, when a user asks an Amazon Echo smart speaker "Alexa, what is the weather today?", the user's audio stream is captured by the smart speaker and transmitted to the cloud for speech recognition, text understanding, and question answering.

However, offloading computation to the cloud has its own downside. Because of the consolidation needed to achieve the economy of scale, today's datacenters are "far" from tier-3 devices. The latency, throughput, and cost of wide-area network (WAN) significantly limit the amount of applications that can benefit from computation offloading. Even worse, it is the logical distance in the network that matters rather than the physical distance. Routing decisions in today's Internet are made locally and are based on business agreements, resulting in suboptimal solutions. For example, using traceroute, we determine that a LTE packet originating from a smartphone on the campus of Carnegie Mellon University (CMU) in Pittsburgh to a nearby server actually traverses to Philedelphia, a city several hundreds miles away. This is because Philidephia has the nearest peering point of the particular commercial LTE network in use to the public Internet. In 2010, Li et al. [9] report that the average round trip time (RTT) from 260 global vantage points to their optimal Amazon EC2 instances is 74 ms. In addition to long network delay, the high network fan-in of datacenters means its aggregation network need to carry significant amount of traffic. As the number of tier-3 devices is expected to grow exponentially, these network links face significant challenges to handle the ever-increasing volume of ingress traffic.

To counter these problems, edge computing, shown as the tier-2 in Figure 2.1, is proposed. Cloudlets at tier-2 creates the illusion of bringing Tier-1 "closer". They are featured by their
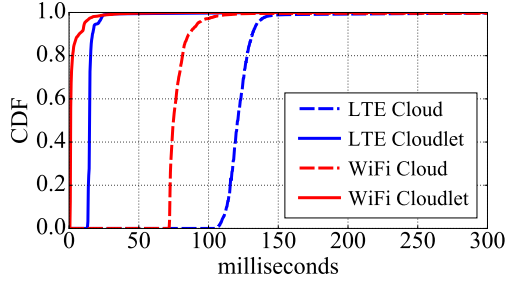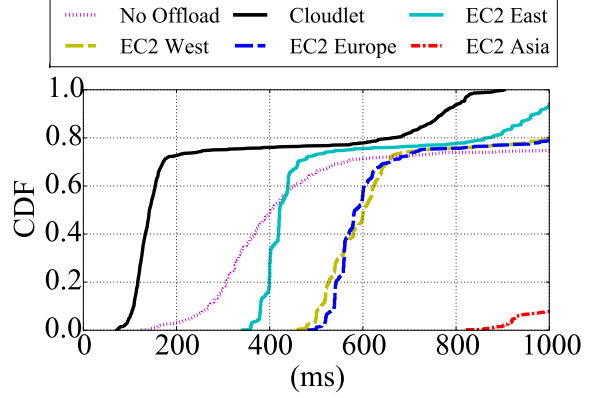
Figure 2.2: CDF of pinging RTTs

Figure 2.3: FACE Response Time over LTE

network proximity to tier-3 devices and their significantly larger compute and storage compared to tier-3 devices. While tier-3 devices typically run on battery and are optimized for low energy consumption, saving energy is not a major concern for tier-2 as they are either plugged into the electric grid or powered by other sources of energy (e.g. fuels in a vehicle). Cloudlets serve two purposes in the tiered model. First, they provide infrastructure for compute-intensive and latency-sensitive applications for tier-3. Wearable cognitive assistance is an examplar of these applications. Second, by processing data closer to the source of the content, it reduces the excessive traffic going into tier-1 datacenters. Figure 2.2 shows the RTT comparison of PING to the cloud and the cloudlet over WiFi and LTE. Cloudlet's RTT is on average 80 to 100ms shorter than its counterpart to the cloud. Figure 2.3 shows the impact of network latency on an application that recognizes faces. Three offloading scenarios are considered: offloading to the cloud, offloading to the cloudlet, and no offloading. The data transmitted are images captured by a smartphone. As we can see, the limited bandwidth of the cellular network further worsen the response time when offloading to the cloud. In fact, for this particular application, even local execution outperforms offloading to the nearest cloud due to network delay. The optimal computational offload location is cloudlet, whose median response time is more than 200ms faster than local execution and about 250 ms faster than the nearest cloud.

The low-latency and high-bandwidth compute infrastructure provided by cloudlets is an indispensible foundation for latency-sensitive and compute-intensive wearable cognitive assistance. Cloudlet also poses unique challenges for scalability as resources are a lot more limited compared to datacenters. How to scale WCAs to many users using cloudlets is the key question this thesis set out to investigate.

2.2 Gabriel Platform

The Gabriel platform [3, 8], shown in Figure 2.4, is the first implementation of WCA and provides an application framework to simplify development. The Gabriel front-end on a wearable device performs preprocessing of sensor data (e.g., compression and encoding), which it streams
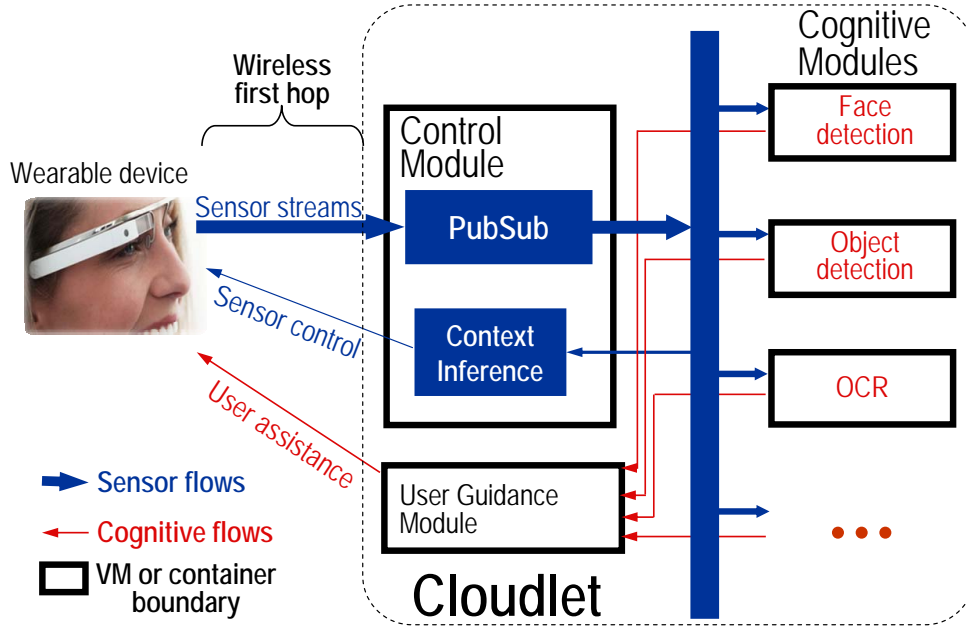
(Source: Chen et al [3])

Figure 2.4: Gabriel Platform

over a wireless network to a cloudlet. The Gabriel back-end on the cloudlet has a modular structure. The *control module* is the focal point for all interactions with the wearable device. A publish-subscribe (PubSub) mechanism decodes and distributes the incoming sensor streams to multiple *cognitive modules* (e.g., task-specific computer vision algorithms) for concurrent processing. Cognitive module outputs are integrated by a task-specific *user guidance module* that performs higher-level cognitive processing such as inferring task state, detecting errors, and generating guidance in one or more modalities (e.g., audio, video, text, etc.).

The original Gabriel platform was built with a single user in mind, and did not have mechanisms to share cloudlet resources in a controlled manner. It did, however, have a token-based transmission mechanism. This limited a client to only a small number of outstanding operations, thereby offering a simple form of rate adaptation to processing or network bottlenecks. We have retained this token mechanism in our system, described in the rest of this paper. In addition, we have extended Gabriel with new mechanisms to handle multitenancy, perform resource allocation, and support application-aware adaptation. We refer to the two versions of the platform as "Original Gabriel" and "Scalable Gabriel."

2.3   Example Gabriel Applications

Many applications have been built on top of the Gabriel platform. Recent papers [3] [2] describe these applications, along with detailed analysis of their end-to-end latency. For example, the LEGO application guides a user to construct a Lego model, step by step, continuously monitoring

the task with computer vision, and providing instructions when it has detected that the user has completed a step. POOL assists a user in aiming a pool cue stick. PING PONG suggests hitting a ball to the left or right to win a rally in table tennis. FACE recognizes a face that has appeared in a scene, searches the user's personal database, and whispers the person's name. IKEA helps a user to assemble an IKEA lamp step by step.

These applications run on multiple wearable devices such as Google Glass, Microsoft HoloLens, Vuzix Glass, and ODG R7. At a high level, the cloudlet workflows of these applications are similar, and consist of two major phases. The first phase uses computer vision to extract a symbolic, idealized representation of the state of the task, accounting for real-world variations in lighting, viewpoint, etc. The second phase operates on the symbolic representation, implements the logic of the task at hand, and occasionally generates guidance for the user. In most WCA applications, the first phase is far more compute intensive than the second phase.

Chapter 3

Conclusion and Future Work

Bibliography

[1] Vannevar Bush and Vannevar Bush. As we may think. *Resonance*, 5(11), 1945. 1

[2] Zhuo Chen. *An Application Platform for Wearable Cognitive Assistance*. PhD thesis, Carnegie Mellon University, 2018. 1, 2.3

[3] Zhuo Chen, Wenlu Hu, Junjue Wang, Siyan Zhao, Brandon Amos, Guanhang Wu, Kiryong Ha, Khalid Elgazzar, Padmanabhan Pillai, Roberta Klatzky, Dan Siewiorek, and Mahadev 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. 1, 2.2, **??**, 2.3

[4] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 17–24. ACM, 2000. 1

[5] Nigel Davies, Keith Mitchell, Keith Cheverst, and Gordon Blair. Developing a context sensitive tourist guide. In *1st Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1*, volume 1, 1998. 1

[6] Zak Doffman. Battlefield 2.0: How Edge Artificial Intelligence is Pitting Man Against Machine. *Forbes*, November 2018. 1

[7] Jason Flinn and Mahadev Satyanarayanan. Energy-aware Adaptation for Mobile Applications. In *Proceedings of the Seventeenth ACM Symposium on Operating systems Principles*, Charleston, SC, 1999. 2.1

[8] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards Wearable Cognitive Assistance. In *Proceedings of ACM MobiSys*, 2014. 1, 2.2

[9] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2010. 2.1

[10] Jack M Loomis, Reginald G Golledge, Roberta L Klatzky, Jon M Speigle, and Jerome Tietz. Personal guidance system for the visually impaired. In *Proceedings of the first annual ACM conference on Assistive technologies*, pages 85–91. ACM, 1994. 1

[11] Jack M Loomis, Reginald G Golledge, and Roberta L Klatzky. Navigation system for the blind: Auditory display modes and guidance. *Presence*, 7(2):193–203, 1998. 1

[12] Brian D. Noble, M. Satyanarayanan, Dushyanth Narayanan, J. Eric Tilton, Jason Flinn, and Kevin R. Walker. Agile Application-Aware Adaptation for Mobility. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, Saint-Malo, France, October 1997. 2.1

[13] John Oakley. Intelligent Cognitive Assistants (ICA) Workshop Summary and Research Needs. `https://www.nsf.gov/crssprgm/nano/reports/ICA2_Workshop_Report_2018.pdf`, February 2018. 1

[14] Tiernan Ray. An Angel on Your Shoulder: Who Will Build A.I.? *Barron's*, February 2018. 1

[15] Mahadev Satyanarayanan. Augmenting Cognition. *IEEE Pervasive Computing*, 3(2), April-June 2004. 1

[16] Mahadev Satyanarayanan and Nigel Davies. Augmenting Cognition through Edge Computing. *IEEE Computer*, 52(7), July 2019. 1

[17] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4), October-December 2009. 2.1

[18] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, pages 14–23, 2009. 1

[19] Mahadev Satyanarayanan, Wei Gao, and Brandon Lucia. The computing landscape of the 21st century. In *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pages 45–50. ACM, 2019. 2.1

[20] Mahadev Satyanarayanan, Guenter Klas, Marco Silva, and Simone Mangiante. The Seminal Role of Edge-Native Applications. In *Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE)*, Milan, Italy, July 2019. 1

[21] Asim Smailagic and Daniel Siewiorek. Application design for wearable and context-aware computers. *IEEE Pervasive Computing*, 1(4):20–29, 2002. 1

[22] Asim Smailagic and Daniel P Siewiorek. A case study in embedded-system design: The vuman 2 wearable computer. *IEEE Design & Test of Computers*, 10(3):56–67, 1993. 1

[23] Asim Smailagic, Daniel P Siewiorek, Richard Martin, and John Stivoric. Very rapid prototyping of wearable computers: A case study of vuman 3 custom versus off-the-shelf design methodologies. *Design Automation for Embedded Systems*, 3(2-3):219–232, 1998. 1

[24] Sherry Stokes. New Center Headquartered at Carnegie Mellon Will Build Smarter Networks To Connect Edge Devices to the Cloud. `https://www.cmu.edu/news/stories/archives/2018/january/conix-research-center.html`, January 2018. 1