Privacy Mediators: Helping IoT Cross the Chasm

Nigel Davies[1], Nina Taft[2], Mahadev Satyanarayanan[3], Sarah Clinch[1], Brandon Amos[3]
[1]Lancaster University, [2]Google, [3]Carnegie Mellon University

ABSTRACT

Unease over data privacy will retard consumer acceptance of IoT deployments. The primary source of discomfort is a lack of user control over raw data that is streamed directly from sensors to the cloud. This is a direct consequence of the over-centralization of today's cloud-based IoT hub designs. We propose a solution that interposes a locally-controlled software component called a *privacy mediator* on every raw sensor stream. Each mediator is in the same administrative domain as the sensors whose data is being collected, and dynamically enforces the current privacy policies of the owners of the sensors or mobile users within the domain. This solution necessitates a logical point of presence for mediators within the administrative boundaries of each organization. Such points of presence are provided by *cloudlets,* which are small locally-administered data centers at the edge of the Internet that can support code mobility. The use of cloudlet-based mediators aligns well with natural personal and organizational boundaries of trust and responsibility.

1. Introduction

In "Crossing the Chasm" [16], Geoffrey Moore warns of a large discontinuity awaiting every new technology as it tries to expand from a small user base of "Innovators" and "Early Adoptors" (Figure 1). In contrast to those early enthusiasts, mainstream users are clear-eyed about the shortcomings of the new technology and seek a net win. Reducing the negatives will increase the chances of success.

The Internet of Things (IoT) is now approaching this chasm, as public awareness of privacy risks grow. In their June 2015 report on consumer perceptions of privacy in IoT [11], Groopman et al state that "Consumers are highly anxious about companies sharing their data: 78% of consumers are highly concerned about companies selling their data to third parties." They also state that "While older generations show higher concern, strong discomfort with the use and sale of connected device data is pervasive across all age groups, including millennials." A January 2015 report [10] by the U.S. Federal Trade Commission notes that "... perceived risks to privacy and security, even if not realized, could undermine the consumer confidence necessary for the technologies to meet their full potential, and may result in less widespread adoption." A June 2015 blog entry [25] notes that "TelecomTV also recently reported a marked slowdown at the fluffy end of the IoT market – people have already had enough (it seems) of home IoT gadgets, so those privacy worries might already be hitting hard." Want et al [29] identify privacy and security as major concerns in IoT.

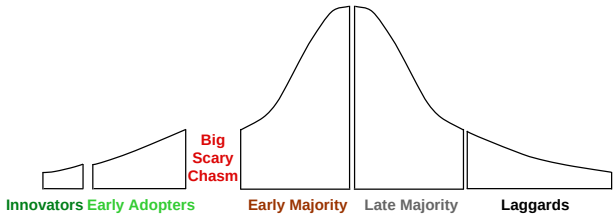

Figure 1: Technology Adoption Chasm (Adapted from Moore [16])

In this position paper, we put forth the view that *concern over data privacy arising from the over-centralization of IoT systems* is a critical obstacle to their growth. There is growing reluctance to expose raw sensor data to a cloud-based IoT entity. IoT deployments today are typically in silos within organisations, or in niche vertical markets. The vision of a rich ecosystem in which shared data is leveraged by a wide range of new applications is yet to be realized.

Our solution is a plug-in architecture with trusted software modules called *privacy mediators* inserted into the data distribution pipeline. A privacy mediator (or, just "mediator") performs data redaction and privacy policy enforcement *before data is released from the user's direct control.* Its platform integrity is ensured by execution on a *cloudlet* [22] in the trust domain of the data owner. This approach delivers a scalable and secure solution at the edge of the cloud, and aligns well with natural organizational boundaries of trust and responsibility. It also scales well to deployments of sensors that have high data rates (e.g., video cameras).

We make the following contributions in this position paper. First, we enunciate an important design principle: namely, that users should have the first option to control the fidelity and distribution of their data. Second, to implement this design principle, we propose an architectural framework that offers a rich set of privacy controls. In our plugin architecture, a small set of trusted third parties (privacy experts) provide the mediation code, thereby reducing the privacy burden on third party app developers. Third, we propose the use of cloudlets (rather than the cloud) to ensure the platform integrity of mediators in the user's eyes.

2. Privacy Control Requirements

Users typically develop a keen sense of what they want from a specific technology only after they have had experience using it. However, in surveys users repeatedly make comments along the lines of "I should get to decide how

HotMobile '16 February 26-27, 2016, St. Augustine, FL, USA

ACM ISBN 978-1-4503-4145-5/16/02.
DOI: http://dx.doi.org/10.1145/2873587.2873600

much of my data a service or application gets to see" and "I should get to decide who my data is shared with" [11]. Users have major angst about continuous ongoing monitoring [25] in which sensors collect measurements at small time scales (such as every minute) and can store seemingly infinite sensor history in the cloud.

Sensors are also becoming increasingly sophisticated. In the future, many of the "things" in the Internet of Things (IoT) will be video cameras. In 2013, it was estimated that there was one surveillance camera for every 11 people in the UK [5]. The report of the 2013 *NSF Workshop on Future Directions in Wireless Networking* [4] predicts that "It will soon be possible to find a camera on every human body, in every room, on every street, and in every vehicle." Such omnipresent video recorders raise significant privacy concerns. Similar concerns have already been expressed by consumers about audio recorders found in smart TVs [20], smartphones, cars, connected toys [21] and game devices. The capture of video and audio streams in private spaces without informed consent can catch consumers unaware, and lead to reputation damage when stories hit the press [20, 21].

While the IoT privacy landscape is complex, a simple principle can serve as a touchstone: *users should be able to control the release of their own data.* This translates into the following privacy control requirements for IoT.

Deletion and denaturing: Users want clear *deletion* capabilities: they need to be able to see (or hear) their data and have the option to delete segments of it. In situations where outright deletion is not required users may still wish to be able to *denature* the data: that is, obscure or modify its sensitive aspects so that it is safe to release to the outside world. For example, faces in images and videos can be blurred[24, 27], and sensor readings coarsely aggregated or omitted at certain times of day or night. Denaturing audio and video is important in many scenarios, especially those involving vulnerable participants (e.g. events involving children, political protests in totalitarian regimes).

Summarization: Providing simple summaries of data for scalar sensors is also possible. For example, a user with a watch sensor might prefer to release only the maximum and minimum heartbeat counts for a day, rather than per-minute measurements. Similarly some users may prefer to release daily or weekly totals of building energy usage rather than per-minute measurements. These are examples of *temporal summaries.* Another kind of aggregation involves *spatial summaries* such as releasing location data at the zip code level rather than raw GPS readings.

Inference: Users want control over *how* their data is used. For example, if temperature or light sensors are used to derive room occupancy as a *virtual sensor* [30], the occupants may want control over the latter sensor too. This concept of a derived virtual sensor applies to many sensors that may not appear too intrusive on their own. However regular readings from such sensors could provide significant insights into the behaviour of the occupants of the house such as their waking times, and levels of physical mobility. Over time, as these virtual sensors get exposed (via the media), users may demand control over them too.

Anonymization: Users may wish to submit data for a societal good, but may prefer to do so anonymously. Examples include data for medical research, and crowd-sourced air quality measurements from smartphones [31] used to monitor specific neighborhoods. A privacy architecture should therefore provide a capability for users to do *anonymization.*

Mobility: Users may also wish to control the flow of data from sensors they briefly encounter in the course of their daily life, for example, in a meeting room or a public space. While not requiring new data filtering, supporting this user mobility places significant demands on any underlying architecture.

Ease of Use: Studies have shown that users often don't understand privacy controls. Although the explicit functionality listed above is needed, users are unlikely to be offered choices defined as above. Instead they would need to be shown a semantic representation of what releasing data at a given granularity could mean. For example, rather than ask users about time granularity for their temperature data, they could be asked if they agree for "room occupancy" to be computed. If they deny this option, data might only be released in a highly aggregated form (e.g. daily total). At the same time, it important not to overwhelm users with too many choices. We further discuss these issues and promising recent efforts in Section 3.5. We believe that it is important to solve the problem of making it easy for users to express and to enforce the controls they need and want.

3. Architectural Approach

The architectural solution we propose is shown in Figure 2. At its heart are *privacy mediators* – pieces of software that run on users' local cloudlets. A mediator is the first point of contact for all data produced by an IoT sensor. It is the mechanism that enforces the privacy policy specified for that sensor. Enforcement occurs in the user's own trusted domain. We describe the components of this architecture in the sections below.

3.1 Cloudlets

Key to our architectural approach is the use of cloudlets: small data centers located at the edge of the Internet, in close proximity to associated sensors and mobile devices [22, 23]. Cloudlets enable cloud services to be virtualised and then instantiated close to their point of use, rather than in the distant cloud. Multiple deployment scenarios are possible. In one option, cloudlets are physically installed in homes, schools or small businesses. It may be possible to install a cloudlet on a high-end Wi-Fi access point, or alternatively on a rack-mounted computer in a wiring closet. Performance studies need to be done to explore these options, but as we show in Section 5.2, a high-end laptop can suffice even for some demanding use cases. An alternate deployment option is for entities such as local telephone companies or cloud service providers to host cloudlets on behalf of home owners. Regardless of deployment model, a cloudlet is always logically within the trust domain of the end user.

3.2 Privacy Mediators

Mediators implement the various types of data privacy controls described in Section 2, and are an integral part of the processing pipeline for IoT sensor data prior to release. We expect mediators to be far more diverse and powerful than the types of simple reverse firewalls and outbound filtering typically deployed at the network edge of many large organizations. Mediators may be specific to a single class of IoT sensor (e.g. a temperature sensor) or may be designed to operate over data produced by many different IoT sensors.

Since sensors may produce data in proprietary formats, we

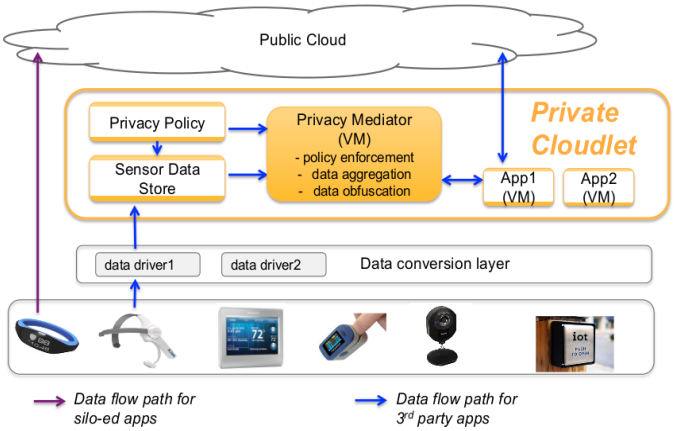

Figure 2: Privacy Architecture

expect that many sensors will require a conversion layer prior to mediation. In our architecture, sensor developers can provide *sensor drivers* (analogous to printer drivers) that convert data into standard or common formats.

Users are able to create policies that control the routing of sensor data to mediators and the configuration of individual mediators. In addition to filtering outbound data, we anticipate that in many cases it will be necessary to store local sensor data in order to perform mediation and access control. The granularity at which data is stored can also be determined by user policy and access to this storage is managed by our privacy policy component. By adhering to the good design principle of separation between policy and mechanism we gain flexibility, such as having mechanisms (e.g. video denaturing) being allowed for some apps or services, but not others.

Since cloudlets are small data centers, virtual machine (VM) encapsulation of a mediator is expected to be the norm. However, in a high-trust deployment, a lighter-weight container such as Docker may be used instead of a VM for encapsulation. The tradeoff is one of increased memory footprint and processor overhead versus superior isolation, safety and smaller attack surface. We expect most mediators to run on a cloudlet within a user's domain. However, our privacy architecture enables dynamic instantiation of mediators on other cloudlets for users that are on the move and need such functionality outside of their usual home or workplace. The use of cloudlets also enables post-mediator application-specific preprocessing of sensor data. This will be important in solving the scalability problem that metro-area networks will face when their input data rates reach excessive volumes due to the proliferation of video capture devices. The combination of mediators and cloudlets thus enables a complete solution to be deployed close to the user, with the attendant benefits in terms of trust, privacy, performance and scalability through reduced latency and bandwidth demands.

3.3 Mediators and Trust

An important design consideration is how mediators and their associated device drivers are produced and deployed. If mediators are produced by the same organisations that supply the sensors and associated cloud services then there is little for the user to gain in terms of privacy guarantees – they would still have to place total trust in the integrity of these organisations' solutions. As a result we expect that mediators will largely be developed by independent third-parties in much the same way as virus checking software is produced today. While some may be proprietary we also expect a significant number of open source mediators to emerge that will benefit from rigorous inspection by the community. Such mediators could obtain trusted reputations by subjecting their code to organizations that give out certifications or seals of approval after inspecting and evaluating a product.

Thus a significant advantage of our architecture is that it could help spur the rich ecosystem vision in which numerous third party developers build applications on top of sensors deployed in the home. One challenge facing such developers is the need to carefully manage user data and try to minimize the occurrence of privacy incidents. Privacy incidents can occur for a variety of reasons, including leakage of private data when a service provider is hacked [28], rogue employees, one company selling data to another who in turn exposes it, communications eavesdropping when insufficient encryption is used [19], or if data gets subpoenaed. Privacy incidents can also occur when a company's transparency in their privacy policy poorly communicates what data is collected [20].

Maintaining up to date security and privacy best practices is challenging, as it requires specific skills and constant vigilance. Some companies achieve this but others do not. It thus seems prudent for less experienced companies to leave part of the privacy responsibility to those who make it a priority and obtain certifications of trust. Our approach creates a data processing pipeline that sits between third party applications and the raw sensor data. Applications obtain data by interacting with mediators via APIs. A privacy cloudlet provides a solution offering privacy functionality that developers could leverage - meaning that they wouldn't need to implement functionality such as denaturing, data aggregation, and policy enforcement, themselves.

In this model, users need to place their trust in a limited number of mediators and the privacy cloudlet service, rather than the hundreds of applications and services they are likely to use. In essence, it reduces the privacy risk surface by limiting the number of players that need to be trusted.

By interacting with user sensor data via the cloudlet, third party applications reduce their own risk and would not have any data that the users has not explicitly consented to release. Note that existing privacy policies do *not* achieve the same effect in terms of obtaining user consent since users are only given the choice to accept or decline the sharing of a particular sensor's data, i.e. today they are not given the choice to control the granularity of the data released.

3.4 Data Storage

In keeping with our basic philosophy of minimizing the threat surface for IoT data we expect that each mediator will maintain its own data buffer. Our architecture does not provide mediators with access to all IoT data on a cloudlet. Rather, mediators are connected to one or more incoming sensor streams and it is only this data that they can access. In this way rogue mediators can only compromise data that have been explicitly granted access to rather than potentially exposing a wider range of a user's IoT data.

In a system that releases only summarized sensor data, the corresponding raw data could be deleted immediately. Alternatively, it could be buffered in the cloudlet for a limited period in case it proves helpful later, e.g., video footage

in a home could be useful in identifying unlawful intrusion. Simoens et al [27] show how a good balance can be achieved between privacy and performance by encrypting and storing raw video sensor data using a randomly-generated private key that is only present within the VM instance (i.e., mediator) for that sensor. Encrypted data can be decrypted on demand, with proper authorization, by the mediator.

3.5 User Policies

A key consideration in the design of our system is how users express their IoT privacy policies in a form that can be enacted using mediators. Specifying such policies has been subject to extensive UbiComp research, with many approaches having a basis in policy languages designed for managing privacy in the web (e.g. P3P and APPEL).

Myles et al. [18] explored location data management – applications wishing to access user locations submit a request that also includes a privacy policy (defined in APPEL), user-registered validators then determine if the requested information can be made available and if any transformations are needed to reduce the granularity of the data. Another policy language, Rei [13], defined four policy object types: rights (permissions an entity has to complete a specified action), prohibitions (explicit records that indicate that an entity cannot complete a specified action), obligations (actions an entity must perform), and dispensations (waivers that excuse an entity from obligations). Since [13] was mainly used for security policies, and [18] was applied only to location data, these approaches need to be explored to understand their extensibility to a broader use for privacy in IoT.

Extending policy languages for privacy in the IoT poses several key challenges, due to the scale of IoT sensors and services, and the issue that users typically struggle to engage with privacy policies in other contexts. First, the *large volume of deployed services*, means that engaging users each time they encounter a new service is simply not viable. Having policies defined per sensor rather than per app results in fewer privacy policies than the per application model we have today on smartphones. However one policy per sensor will still be too much; thus new approaches such as defining policies by class of sensor, hierarchically or other groupings of sensors, needs to be explored. Second, users may want to have *recipient-specific preferences*. A user's policy for sensors is likely to vary depending on the data recipient, and the perceived value or risk of sharing with that recipient. For example, a home owner may be willing to share detailed data with their energy supplier (for the purpose of improving the provided service or managing billing), but may only wish to share summarizations with a local government or third-party application.

To overcome these challenges and abstract over specific IoT devices and services we suggest the use of high-level privacy goals that can then be translated into mediator requests. Based on overall privacy and service goals, we propose that users maintain general profiles consisting of reusable policies that apply to classes of application or device.

However, expressing these high-level goals and profiles still requires input from users. Current privacy literature suggests two approaches for encouraging users to maintain their privacy goals and profiles. We need to determine a set of smart default privacy profiles that capture a range of opinions about privacy. In the context of permission management for Android applications, Liu et al. [15] showed that it is possible to find a small set of (4-6) default privacy profiles that capture the preferences of many users and allow for simple customizations. Each default profile in [15] contains a list of permissions typically declined, along with those typically accepted. Employing smart default profiles means that users only need to engage when something outside of their base profile occurs. A second approach is to use an active privacy assistant [1] that advises users what to do only when things they might be concerned about arise. This work showed that when users are nudged with one privacy question per day, 60% of users were responsive and actively changed permission settings. Clearly much work remains, but the initial success of these approaches is encouraging.

4. Business Models

Our focus so far has been largely on home scenarios deploying third party applications and services. Clearly there are other business models for home services based on first party applications in which users buy a sensor directly from the service provider for a particular service - such as Opower and Fitbit. Such services can co-exist in a home deploying a private cloudlet by simply bypassing the cloudlet (as depicted in Figure 2 by the blue arrow on the left).

Our architecture is also well suited to meet the needs of businesses in which privacy policy may be delegated to administrators within a larger organization. In schools and university settings, privacy policy would likely be set by an administrator or a particular teacher. For example, elementary schools with video cameras in the classroom, might elect to release videos with the faces of children blurred - this could be useful if a school district wanted to evaluate the effectiveness of teaching tools while simultaneously protecting the privacy of individual children. Schools employ third party applications to supplement teaching materials and our architecture makes it easy for third party applications to be compliant with school privacy policies around data use.

There are many businesses from small to large that will be managing IoT sensors in buildings of a workplace. Different companies will have different sensitivities and rules. For example, medical establishments, financial establishments and law firms have different legal requirements around patient and client data. Lawyers need to be sure that accidental audio recording is not moved to the wrong place when interacting with clients. Dueling needs of hospitals are well known - they must protect patient privacy, but at the same time would like to release summarized data to improve medical research. They could instantiate our privacy cloudlet by having the storage and mediator capabilities run on a hospital's IT infrastructure and have research efforts get data via interaction with mediators.

While our architecture is able to support a range of existing IoT business models we note that it may also stimulate entirely new areas of economic activity such as the supply and validation of mediators and associated drivers.

5. Challenging Use Cases

5.1 Human Augmentation

An area of intense research interest at present is the use of pervasive technologies to augment human capabilities such as memory [9]. Cameras, microphones and other environmental sensors can be coupled with wearable devices such

as lifelogging cameras to provide rich datasets relating to a user's experiences. This data can potentially be processed and then used to cue recall of memories as required. Applications range from behaviour change and increased learning capacity to support for failing memories. To understand the challenges of such systems the authors conducted the RECALL experiment in which twenty researchers, wearing a range of lifelogging devices, spent two days in an instrumented hostel to capture a test dataset for memory augmentation [7]. Fixed infrastructure cameras throughout public areas of the hostel recorded a continuous video stream and participants were equipped with wearables such as smartphones, GoPros, Narrative Clips, SenseCams and DSLR cameras. Over 280GB of data was captured including 42,959 images and 248.15 hours of video and location data.

Clearly such a scenario raises a number of significant privacy concerns with potentially very large amounts of data being captured and users interacting with a wide range of sensors in the infrastructure [9]. However, the proposed architecture begins to provide an insight into how such systems could emerge while providing user control over privacy. For example, mediators running on a local cloudlet could redact a video feed to a few still images that are used as the basis of memory cues. To support memory augmentation more generally, mediators could eventually become complex pieces of software that locally determine which few elements of the sensor data needed to be exported – enabling most of the data to remain private within the user's local cloudlet.

While our architecture offers an obvious solution for domestic and work spaces, scenarios such as memory augmentation highlight the challenge of mobile users that wish to control the capture of data relating to them as they move between instrumented spaces. Our approach of using cloudlets to support code mobility provides a robust, scalable and secure means of allowing users to dynamically instantiate mediators into spaces that they temporarily access. Clearly trust relationships still need to exist between users and the spaces themselves but this is always likely to be the case (it is not possible to prevent, for example, a space owner deploying hidden cameras). However, where such trust relationships exist cloudlets would provide a natural way to support the dynamic instantiation of one or more mediators for a user that has temporarily appropriated a physical space such as a meeting room.

5.2 Omnipresent Video

Today, most video is stored in silos close to the point of capture. In the future, we envision many use cases in which analysis of multiple video streams and fusion of extracted information offers powerful benefits to users [24]. This requires today's isolated video cameras to be integrated into an IoT framework, posing challenges for scalability as well as privacy. The high cumulative data rate of incoming videos from many cameras is a key scalability challenge.

Our privacy architecture can also solve this scalability challenge by running video analytics on cloudlets. Simoens et al [27] have shown that denaturing and video analytics at throughput acceptable for a typical home are feasible on cloudlets of modest computational power. They recommend sampling an input video stream at a lower rate than the capture rate, and to perform video analytics and denaturing only on the sampled frames. They also suggest that raw video be stored encrypted on the cloudlet, and only decrypted and analyzed if an explicit need to examine that video is identified. All this functionality is supported by our architecture. Shipping the extracted tags and meta-data to the cloud only requires modest bandwidth.

Denaturing has to strike a balance between privacy and value. At one extreme is a blank video: perfect privacy, but zero value. At the other extreme is the original video at its capture resolution and frame rate. This has the highest value for potential customers, but also incurs the highest exposure of privacy. Where to strike the balance is a difficult question that is best answered individually, by each user. This decision will most probably be context-sensitive. One example is to blur all faces in an image; this only requires face detection, which is a standard capability in image processing software today. A more selective privacy policy might only require the faces of certain people to be blurred. That is considerably more difficult to implement, since face recognition is a much harder computer vision problem than face detection. Fortunately, we have been successful in creating an open source implementation of face recognition using deep neural networks called *OpenFace* that provides surprisingly high accuracy [2].

OpenFace can be used as a building block for IoT services running on a cloudlet. For example, when running on a laptop such as Macbook Air, OpenFace can train to learn a new face (e.g. guests in a home) in 10-20 seconds. Subsequent recognition takes about 500 ms in our prototype. Both of these functions can thus be performed without having to offload any video to the cloud. This serves as an illustration that even complex mediators can be implemented to run in private cloudlet architectures.

6. Related Work

Protecting user privacy has been extensively explored in the UbiComp community, e.g. informing users of potential privacy threats [14] and protecting user location data [6], [8]. Our work focuses on the use of privacy mediators. The idea of a rule-based trusted intermediary that controlled the release of location information was described in [18] and a similar, though more general solution was later proposed in [17] in which Personal Data Vaults were used to filter end-users' mobile sensor data before sharing it with content-service providers. Our work differs from these rule-based intermediaries by offering a generalized cloudlet infrastructure for intercepting both mobile and (predominantly) fixed sensor data.

In [3] the use of OSGI for hosting privacy interceptors in smart environments was explored – this approach has many parallels with our architecture but focuses on enforcing privacy policies relating to contextual data when requests for information are received (and thus is similar to [18]) rather than processing outgoing sensor streams. Moreover, the choice of OSGI as an underlying platform is obviously limiting compared to the more general support provided by cloudlets. Commercial firewalls for smart homes are beginning to appear (e.g. http://www.bitdefender.com/box/) but these typically focus on protecting the home from inbound traffic rather than protecting privacy through mediation of outbound traffic from uncompromised devices.

In [26] a decentralized infrastructure for social networking is proposed where each user's configurable"butler" provides fine-grained access control and storage. In contrast, we do not address the needs of social networking, nor focus on dis-

tributed storage, nor prevent the sending of sensor data to the cloud (instead we enable the latter at a user-chosen granularity). Other work in this field includes [12] that investigates on-device sensor abstractions for augmented reality applications to prevent private data from accidentally being leaked from applications having raw sensor data access.

7. Conclusion

In this position paper we addressed the challenge of helping the IoT bridge the impending "chasm" that blocks the path to widespread adoption. We have argued that privacy is a key issue and subsequently proposed an architecture based on an essential design principal, namely that users should maintain overall control of their data and be responsible for managing its release to cloud services. Our architecture provides a framework for addressing privacy requirements in traditional IoT environments and, crucially, through the use of cloudlets and mobile code enables the support of challenging IoT scenarios in the mobile domain including human augmentation and omnipresent video. We have focused herein primarily on home scenarios because that is where the need is most urgent for IoT. In our future work, we plan to study extensions of our architecture for public space IoT applications.

Acknowledgements

This research was partially funded through the National Science Foundation (NSF) under grant number CNS-1518865. Additional support was provided by Intel, Google, Vodafone, Crown Castle, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and should not be attributed to their employers or funding sources.

8. REFERENCES

[1] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. Cranor, and Y. Agarwal. Your Location has been Shared 5,398 Times! A Field Study on Mobile App Privacy Nudging. In *Proc. of ACM CHI*, 2015.

[2] B. Amos. OpenFace: free and open source face recognition with deep neural networks. http://cmusatyalab.github.io/openface/, 2015.

[3] S. A. Bagüés, A. Zeidler, C. F. Valdivielso, and I. R. Matias. Sentry@Home-leveraging the smart home for privacy in pervasive computing. *International Journal of Smart Home*, 1(2):129–145, 2007.

[4] S. Banerjee and D. O. Wu. Final report from the NSF Workshop on Future Directions in Wireless Networking. NSF, November 2013.

[5] D. Barrett. One surveillance camera for every 11 people in Britain, says CCTV survey. *Daily Telegraph*, July 10, 2013.

[6] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, Jan. 2003.

[7] S. Clinch, N. Davies, M. Mikusz, P. Metzger, M. Langheinrich, A. Schmidt, and G. Ward. Collecting shared experiences through lifelogging: Lessons learned. *Pervasive Computing, IEEE*, 15(1), 2016.

[8] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proc. of ACM MobiSys*, pages 211–224, 2008.

[9] N. Davies, A. Friday, S. Clinch, C. Sas, M. Langheinrich, G. Ward, and A. Schmidt. Security and privacy implications of pervasive memory augmentation. *Pervasive Computing, IEEE*, 14(1), 2015.

[10] FTC Staff. Internet of Things: Privacy and Security in a Connected World. Technical report, Federal Trade Commission, January 2015.

[11] J. Groopman and S. Etlinger. Consumer Perceptions of Privacy in the Internet of Things: What Brands Can Learn from a Concerned Citizenry. Technical report, Altimeter Group, June 2015.

[12] S. Jana, D. Molnar, A. Moshchuk, A. M. Dunn, B. Livshits, H. J. Wang, and E. Ofek. Enabling fine-grained permissions for augmented reality applications with recognizers. In *USENIX Security*, pages 415–430, 2013.

[13] L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *Proc. of IEEE POLICY*, 2003.

[14] M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In *Proceedings of the 4th International Conference on Ubiquitous Computing*, UbiComp '02, pages 237–245, London, UK, UK, 2002. Springer-Verlag.

[15] B. Liu, J. Lin, and N. Sadeh. Reconciling mobile app privacy and usability on smartphones: could user privacy profiles help? In *Proc. of ACM WWW*, 2013.

[16] G. Moore. *Crossing the Chasm*. Harpercollins, 1991.

[17] M. Mun, S. Hao, N. Mishra, K. Shilton, J. Burke, D. Estrin, M. Hansen, and R. Govindan. Personal data vaults: a locus of control for personal data streams. In *Proc. of ACM CoNEXT*, page 17, 2010.

[18] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *Pervasive Computing, IEEE*, 2(1), 2003.

[19] At&t hacker "weev" sentenced to 41 months in prison, after obtaining the email addresses of 100,000+ ipad users. https://nakedsecurity.sophos.com/2013/03/19/att-hacker-weev-prison/, 2013.

[20] http://bits.blogs.nytimes.com/2015/02/10/samsung-tweaks-television-policy-over-privacy-concerns/.

[21] C. News. Talking Barbie is too creepy for some parents. http://money.cnn.com/2015/03/11/news/companies/creepy-hello-barbie/, March 2015.

[22] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE*, 8(4), 2009.

[23] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani. An Open Ecosystem for Mobile-Cloud Convergence. *IEEE Communications Magazine*, (3), March 2015.

[24] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos. Edge analytics in the internet of things. *Pervasive Computing, IEEE*, 14(2), 2015.

[25] I. Scales. Is IoT going to be squashed because of privacy concerns? http://www.telecomtv.com/articles/iot/is-iot-going-to-be-squashed-because-of-privacy-concerns-12647/, June 2015.

[26] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam. Prpl: a decentralized social networking infrastructure. In *Proc. of ACM MCS '10*, page 8, 2010.

[27] 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*, 2013.

[28] Health insurer didn't encrypt data in theft. http://www.wsj.com/articles/investigators-eye-china-in-anthem-hack-1423167560, 2015.

[29] R. Want, B. N. Schilit, and S. Jenson. Enabling the Internet of Things. *IEEE Computer*, 48(1), Jan 2015.

[30] T. Weng, A. Nwokafor, and Y. Agarwal. Buildingdepot 2.0: An integrated management system for building analysis and control. In *Proc. of ACM BuildSys*, 2013.

[31] W. Willet, P. Aoki, N. Kumar, S. Subramanian, and A. Woodruff. Common sense community: Scaffolding mobile sensing and analysis for mobile users. In *Pervasive Computing Conference*, 2010.