Personal Data Management with the Databox: What's Inside the Box?

Richard Mortier,
Jianxin Zhao,
Jon Crowcroft,
Liang Wang, Qi Li
The Computer Laboratory,
University of Cambridge, UK
first.last@cl.cam.ac.uk

Hamed Haddadi,
Yousef Amar
School of EECS
QMUL, UK
first.last@qmul.ac.uk

Andy Crabtree,
James Colley,
Tom Lodge,
Anthony Brown,
Derek McAuley,
Chris Greenhalgh
School of Computer Science,
University of Nottingham, UK
first.last@nottingham.ac.uk

ABSTRACT

We are all increasingly the subjects of data collection and processing systems that use data generated both about and by us to provide and optimise a wide range of services. Means for others to collect and process data that concerns each of us – often referred to possessively as "your data" – are only increasing with the long-heralded advent of the Internet of Things just the latest example. As a result, means to enable personal data management is generally recognised as a pressing societal issue.

We have previously proposed that one such means might be realised by the *Databox*, a collection of physical and cloud-hosted software components that provide for an individual data subject to manage, log and audit access to their data by other parties. In this paper we elaborate on this proposal, describing the software architecture we are developing, and the current status of a prototype implementation. We conclude with a brief discussion of Databox's limitations.

Keywords

Personal data management; Edge network services

1. INTRODUCTION

Increased ubiquity of sensing via mobile and IoT devices has caused a surge in personal data generation and use. Alongside this surge, concerns over privacy, trust, and security are becoming increasingly important as different stakeholders attempt to take advantage of such rich data resources: occurrences of breaches of privacy are rising at alarming rates [15]. Tensions in the collection and use of personal data, between the benefits to various analytics applications, the privacy consequences and security risks, and the regulatory complexities of aggregating and processing data in the cloud are a significant barrier to innovation in this space. We have previously proposed that these topics, and the shortcomings of current approaches in this space, are the concern of a new – or at least, newly focused – discipline, Human-Data Interaction (HDI) [19].

In our view the core technical problem in this space is how to build networked services that enable individuals to manage their personal data so that they can permit other parties to make use of it while retaining personal control over such uses and understanding the implications of any data release. As digital data may be copied infinitely at negligible marginal cost and without loss of fidelity, the current common approach of centralising unencrypted personal data into a cloud-hosted service such as Google or Facebook is fundamentally flawed in this regard. Once data is given up to such a service, the data subject can only exercise control over it to the extent that the cloud provider allows them, and they have only social means (e.g., negative publicity campaigns, local per-jurisdiction regulation) to ensure such controls are provided.[1]

Our response is to provide technical means to assist the data subject in managing access to their data by others. As we have previously proposed, the Databox is an open-source personal networked device augmented by cloud-hosted services that collates, curates, and mediates access to our personal data, under the data subject's control [11]. It sits within an ecosystem of networked devices and associated services enabling individuals to manage their data, and to provide other parties with controlled access to their data. Composed of a set of service instances, realised as Docker-managed containers in our current prototype, it enables Cloud-Assisted Networking through the placement of these instances in different locations, from a physical device in the subject's home, to the public cloud, to future envisioned edge-network hosting resources such as smart lampposts and cell-towers.

Databox not only benefits data subjects by providing a regulated and privacy-enhanced communication mechanism between data subjects and data processors. Acting as an agent on behalf of the data subject, it can support queries

CAN'16, December 12 2016, Irvine, CA, USA

ISBN 978-1-4503-4673-3/16/12...$15.00

DOI: http://dx.doi.org/10.1145/3010079.3010082

[1] We follow standard legal terminology and refer to individuals about whom data is collected from *sources* such as IoT sensors or online social network accounts as *data subjects*, and organisations wishing to process data as *data processors*.

over high-resolution personal data that would be difficult for a single company to obtain, permitting richer, more accurate data analytics. It also helps avoid the risks of data breach associated with collecting and curating large, personal datasets which malicious actors have significant incentives to attack and steal.

In this paper we explore the software architecture of the Databox, and describe in more detail how we envisage it being used. We first briefly survey related approaches to supporting capture and processing of personal data (§2). We then set out the software architecture of our current prototype, focusing on the four primary components: store, driver, manager and apps (§3). We then briefly discuss how we envisage data processing and analytics taking place in a world of Databoxes (§4). We finish by describing the status and initial performance of our current prototype (§5) and concluding (§6).

2. PERSONAL DATA MANAGEMENT

Personal data collection for profiling and mining users' interests and relationships is the basis on which online platforms such as Facebook and Google and services such as Apple's Siri and Microsoft's Cortana operate. However, such data collection and profiling exposes the user to privacy leakage even when these communities are anonymous [6]. Simultaneously, these cloud-based services can have only a partial view of each data subject's digital footprint, resulting in inaccuracies and systemic biases in the data they hold, and leading to ever more aggressive data collection strategies.

Building privacy, trust and security into the evolving digital ecosystem is thus broadly recognized as a key societal challenge. Regulatory activities in the US [28], Europe [8] and Japan [26] are complemented by industry initiatives that seek to rebalance the "crisis in trust" [29] occasioned by widespread personal data harvesting. All parties agree that increased accountability and control are key to this challenge. *Accountability* seeks not only to strengthen compliance but also to make the emerging ecosystem more transparent to consumers, while *control* seeks to empower consumers and provide them with the means of actively exercising choice.

Although numerous mechanisms supporting privacy preserving analytics, marketing and advertising have been proposed, e.g., recent studies on analysing network traces using differential privacy [7] and accessing databases while respecting privacy [17, 24], no operational system exists that also gives others visibility into statistics and trends [13, 10, 12]. Rieffel *et al.* [25] propose cryptographic, hierarchical access to data for processing aggregate statistics without decrypting personal data. However this method still requires collection of individual data items and complex yet critical management of many cryptographic keys. Privacy-aware centralised methods such as homomorphic encryption [21] are yet to be deployed in a commercial or consumer system. While these methods are likely to be important in the future, they are not enough alone: they cannot provide accountability and control in isolation.

Numerous recent and current projects and startups have responded to specific problems of aggressive data collection by the online advertising industry[2] through more traditional means. These typically involve production of services called variously *Personal Data Stores*, *Personal Information Management System*, *Vendor Relationship Management*, and similar; examples include Mydex [1] and openPDS [5]. They allow the subject to retain "ownership" of their data and provide it to third parties on demand [23], which offers some degree of accountability and control but only insofar as the service provider can be trusted. Simple notions of "ownership" are also problematic, a point to which we return in (§6).

Our approach, the Databox, is a set of networked service enabling individuals to manage their data, and to provide other parties with controlled access to their data. It gathers data from local and remote sources, from online social networks to IoT sensors; provides for data subjects to inspect data gathered from their data sources, and to effect actuation via IoT devices and similar; enables data processors to discover and request access to subjects with sources of interest; and it supports running applications to provide data processors with specific, limited, logged access to subjects' data.

3. INSIDE THE DATABOX

Given currently available infrastructure, we expect the common-case Databox deployment to be a hybrid of a local physical device (or devices) and cloud-hosted services. The physical instances might come in the form-factor of an augmented home broadband router, and provide several affordances not easily available to a pure cloud-hosted solution: direct physical control (e.g., you can unplug the device and be certain no-one can access your data through it); co-location (e.g., imposing data access policies referring to shared data such as from a smart energy meter might require the physical presence of more than one data subject); proximity (e.g., data access might require that the data subject is physically present at the moment when access is granted); and the intuitiveness of physical interaction (e.g., as with the Homework Router [20], selected policies might be imposed and relaxed by the introduction or removal of a colour-coded USB key to the device). Note that the software architecture of the Databox means that as other cloud-hosting becomes available, such as those hosted in the edge network, in celltowers, in automobiles, in lampposts, the Databox will be able to take advantage of these facilities.

3.1 Threat Model & Design Principles

Before laying out the design principles to which we adhere we must briefly discuss those threats we set out to deal with and those we do not. Specifically, we are not proposing the Databox as a generic solution to problems of online anonymity or security. Rather, it is intended to provide means for individuals to understand and manage the uses made of their personal data, per the core themes of HDI (legibility, agency, and negotiability).

This motivates the approach of Databox: data processing is mediated by and hosted on the Databox itself, even where raw data is held outside the Databox (e.g., your online social network data). This is in stark contrast to the widespread current approach whereby personal data is handed over to some organisation through which it can be managed. By avoiding this, we can ensure detailed and coherent logging of all operations carried out on the subject's personal data,

[2] http://www.technologyreview.com/view/530741/the-murky-world-of-third-party-web-tracking/

as well as tight control over the release of data derived from the processing of your personal data.

As with all edge-cloud solutions, this greatly reduces the attack surface: it is no longer the case that a single misplaced click can release the records of millions of users – instead, millions of clicks are required. We also significantly reduce the core temptation of an honest-but-curious attacker who would otherwise be in the position of holding many users' personal data: as accesses must now be via each individual's Databox, the payoff of simultaneous access to millions of records is reduced.

With that starting point in mind, we attempt to separate concerns, distinguishing data collection from storage from access. In practice this translates to:

All components are clearly separated. Databox follows a micro-services pattern: components are separated at as low an OS layer as possible, and inter-communicate using explicit APIs. This also helps ensure portability between physical device and cloud hosting. Our prototype uses containers to provide this separation; future implementations might provide stronger separation through use of unikernels or virtual machines [2, 18].

Distinct data sources are represented by distinct stores. This helps to ensure that an attacker who gains access to one store does not, de facto, gain access to any other store without performing privilege escalation against the host platform.

Components are disconnected by default. This decreases the likelihood of breach by narrowing the window of opportunity for an attacker. Any network-connected system can be attacked at any time: by ensuring that components that do not need to be connected remain disconnected, the number of components an attacker can attack is reduced to a small number of components that must necessarily be reachable all the time; particular attention can be paid to hardening these.

All control and data flow is logged to enable subsequent audit. This increases the likelihood that, in case of data breach, the source of breach can be traced and the associated risks and costs can be quantified and mitigated. In light of the preceding principle, component inter-connection is thus via purpose-specific bridges that are able to validate the identity of the external party making the connection via (e.g.,) TLS Client Certificates.

3.2 Prototype Architecture

Figure 1 depicts the software architecture of the Databox. There are four core components at the heart of the Databox which we describe next: (*i*) driver, (*ii*) stores, (*iii*) the manager, and (*iv*) apps.

Drivers. Each data source is represented in the Databox via a driver. In common with many device driver models, the driver can be thought of as the coupling of a frontend and backend. The backend is source-dependent, and responsible for all interaction with the source. The frontend is the standardised interface provided by the attached store, through which other components interact with the backend. Drivers read and write (or collect and actuate) in response to invocations made on their store(s). Each driver is responsible for exactly one source; each source has exactly one driver on a given Databox.

Stores. The Databox contains many data stores, either primary or derived, all implementing the same standard

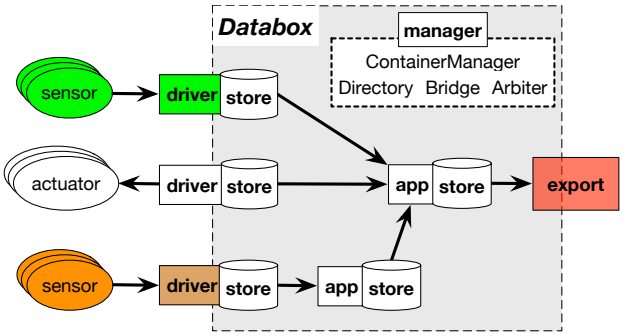

Figure 1: Proposed Databox architecture. Drivers interface external devices (sensors and actuators) into the Databox. All invocations and data flow through stores, which log operations and (potentially) data. Apps consume data from stores and publish results into new stores. The Manager enables access to selected stores by external data processors.

API. Primary stores connect to drivers, and record "raw" data collected directly. All accesses through a store, whether read or write, are logged into a distinct, system provided store, which makes them available for subsequent visualisation and audit. Accesses to stores depend on permission being granted via the Manager.

Manager. A Databox contains a set of management functions: (*i*) *container manager*, managing container instances; (*ii*) *directory*, recording all installed drivers and applications, aiding discovery; (*iii*) *bridge*, providing interconnection between running components; and (*iv*) *arbiter*, responsible for managing interactions between components, and between components and external parties. In particular, the arbiter is responsible for minting and validating tokens permitting external parties to connect to stores, and for enabling connectivity between drivers, applications, derived stores, and external processors.

Apps. Finally, the Databox hosts isolated applications provided by third parties. These perform the actual data processing required to provide specific services. Applications then connect to one or more stores (either primary or derived) and record into derived stores data resulting from computations carried out on the contents of their input stores. Thanks to the logging inherent in each store, all accesses (reads and writes) are recorded separately for later audit.

All these components are open-source, and applications then interact with them over TLS via published APIs. We next discuss the ways we envisage processing taking place across multiple Databoxes.

4. DATA PROCESSING & ANALYTICS

As the primary purpose of the Databox is to support an ecosystem around processing of personal data, applications are a key part of the architecture: how they are discovered, installed and operate.

We envisage that all Databox applications will involve some software component running within the Databox. Specifically, applications provide derived stores to which external

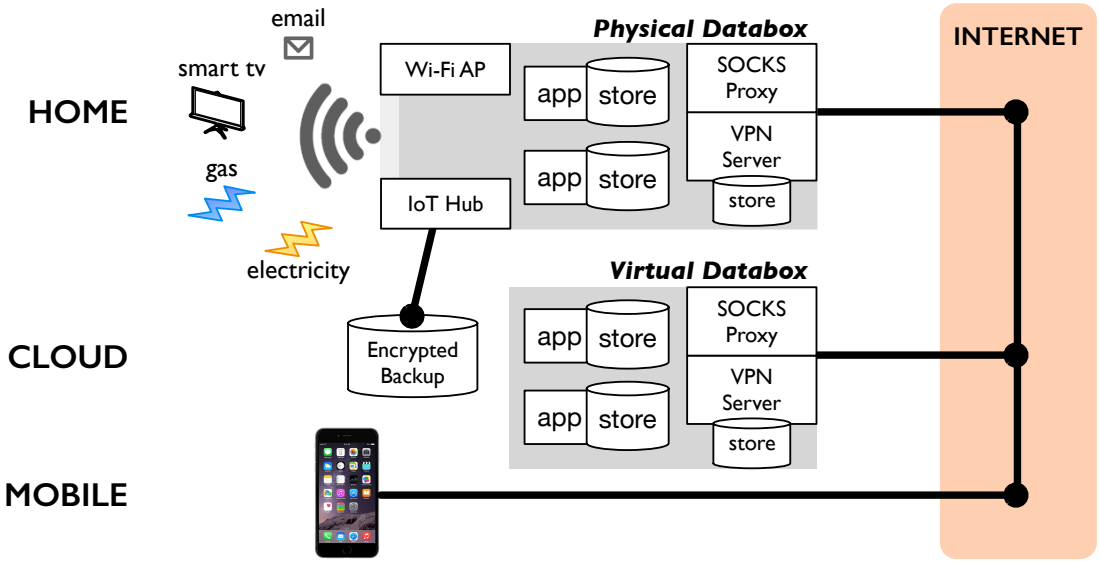

Figure 2: Databox configured to capture an individual's online activity, in all locations, across all devices.

parties (whether data processing organisations or a browser operated by the data subject) can connect. These applications provide accountable entities through which the data subject can ascribe to a data processor behaviour involving use of their data. We envisage two main routes to installation of these components, resulting from successful negotiation between the data subject and processor causing the processor to be given access to the subject's data: subject-driven and processor-driven.

Subject-driven. This model is strongly analogous to current app store models. Each app store is a service to which subjects can connect to discover applications they might wish to install. Apps advertise the data stores that they will consume, the frequency at which they will access those stores, and the computation they will carry out on data from those stores. Apps will be validated and verified by the app store operators, and rated by the community. They will have only limited, approved external communications between their own store and an authenticated client.

Processor-driven. This model inverts the interaction, enabling data processors to discover cohorts of data subjects who have data available to meet the processor's needs.

Independent of the discovery model used, applications may either be limited to 1:1 interactions, or may necessarily involve a larger cohort of subjects making data available. In the former case, the output of the computation is consumed in isolation, either by the data subject or the data processor. In the latter, there is an inherent need for the application to function as a distributed system, with communication taking place between instances as the computation makes progress. This latter case is considerably more complicated so we discuss it briefly next.

Three key challenges of this sort of application present themselves: **scale**, **heterogeneity**, and **dynamics**. These challenges arise due to the fundamental characteristics of the algorithms and deployment environments envisaged. Data processors might use machine learning and model generation algorithms that default to serial computation or, at best, execute in the controlled, high bandwidth, low latency environment of a datacenter. Scaling them across (potentially) millions of Databoxes moves most of these algorithms outside their normal operating regions. The computation resources on which they will be hosted will vary in capacity and connectivity, making scheduling and synchronisation of ongoing computations between instances considerably more complex, Finally, physical Databox instances are likely to have variable and unreliable connectivity which, when coupled with the envisaged scale, almost guarantees that the entire cohort will never be simultaneously available.

Approaches to addressing these challenges that we are exploring include the use of techniques such as delay-tolerant querying, introduced in the Seaweed database [22], where metadata statistics are incorporated into the decisions taken by the system as to when to wait for data to become available and when to give up and return (with appropriate indications) a potentially inaccurate answer; and more flexible control over synchronisation barriers than permitted by Bulk Synchronous Parallel operation (e.g., Stale Synchronous Parallel Parameter Server [14]).

It is possible that other factors inherent in these data and the deployment of Databoxes may also mitigate against some of these problems. For example, the distributed computations may well be highly localised and so might be loosely coupled and require minimal coordination and exchange of data. Coupled with use of aggregation in the computation graph, this might mitigate unreliability and scale, while also providing natural means to support privacy-preserving aggregation.

5. CURRENT STATUS

We are in the process of building a first complete, open-source prototype of the Databox. The hardware platform consists of the Docker Cloud for cloud-hosted components and small form-factor PCs (Intel NUCs and Raspberry Pis) for the locally-hosted components. The software platform is

based around the Docker container management platform, with each software component instantiated and managed as a container. Use of Docker makes support for different platforms and architectures substantially more replicable and straightforward, as well as ensuring a familiar and portable development environment. We are currently pursuing two application domains: (*i*) individual online behaviour across the multiple network-connected devices most of us now use; and (*ii*) domestic Internet of Things across a wide range of sensors and actuators targeting household settings.

In the first case we will capture information about our online interactions with people and companies using a range of devices including tablets, phones, and traditional PCs. This data is often already captured but typically only by other parties, from ISPs to social media companies to retailers – the result is that it is usually opaque to us, and every company forms only a partial and inaccurate picture of our behaviours.

The current prototype, depicted in Figure 2, sets up *drivers* that run IKEv2/IPSec VPN and Socks/HTTP proxy servers, and the drivers then dump captured data into a *store*. Both drivers and stores are implemented as Docker containers, and a further container hosts a webserver that provides configuration instructions, customised to the connecting device, for each of these services. The subject configures their mobile devices to communicate via on or both of these, allowing the subject to log the network activity of their devices and applications.

From initial performance tests we find that latency and throughput are within 80% when VPN and SOCKS proxies are hosted in the cloud, but (as expected) are significantly degraded (throughput reduced by over 70% and latency similarly affected when installed on the low power ARM-based devices). Although this is an unoptimised setup, when coupled with the usual problems of connectivity through home gateways, this seems to support the hybrid approach and suggests that correctly managing heterogeneous resources will be a key challenge.

The second case involves configuring drivers and stores on the Databox to capture data from a range of domestic sensors and to interact with installed actuators. The current prototype already interfaces to a range of off-the-shelf sensors (accelerometers plus CO_2, temperature and humidity detectors), as well as standard smartphone hardware sensors available via iOS and Android. We have also developed a visual programming environment that allows simple creation of IoT apps to run on the Databox.

6. DISCUSSION & CONCLUSIONS

We have briefly presented the software architecture for the Databox, a hybrid locally- and cloud-hosted system for personal data management. While it addresses concerns of privacy and ethics of these data, it does not try simply to *prevent* all such analysis and use by third-parties as not all of this activity is harmful [3, 16]. Rather, it seeks to afford users the possibility to find personal equilibria with sharing and use of their data: simply preventing all access to personal data would fail to take advantage of the many potential benefits that sharing data can produce, whether immediate financial rewards, social benefits through, e.g., participation in friendship groups, or broad societal benefits from the ability to participate in large-scale studies in, e.g., human mobility and activity for use in urban planning, or mental and physical health measures used to set healthcare norms.

The current prototype implementation is a first step to realisation and evaluation of our proposed architecture. Use of Docker containers provides a very convenient way to experiment with the structure of the system in these initial stages. Longer term we are interested in exploring use of unikernels, such as MirageOS [18], to implement some of the very specific and limited functionality software components that are security critical such as those responsible for authenticating and configuring connectivity to applications and drivers, as well as the store, perhaps by building on Irmin [9] libraries to produce a store where all read/write operations are logged.

While we believe that the Databox will be useful in and of itself, the present design is clearly not the final word in personal data management. Although personal data management is generally considered an intensely personal matter [27], it is also inherently social: it is impractical to withdraw from all online activity simply to protect one's privacy [4]. Similarly many households have multiple occupants and data sources may therefore be inherently accountable to several data subjects. What can be done to enable meaningful engagement in the management of such shared personal data is an open challenge at present.

Acknowledgements. This work was funded in part by grants: EU FP7/2007–2013 Grant No. 611001; and EPSRC grants EP/N028260/1, EP/M001636/1 and EP/N028422/1. We would also like to acknowledge constructive feedback from Prof. David Kotz (Dartmouth College).

7. REFERENCES

[1] Mydex. https://data.gov.uk/library/mydex, 2012.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proc. ACM Symposium on Operating Systems Principles (SOSP)*, pages 164–177, Bolton Landing, NY, USA, 2003. ACM.

[3] D. Boyd and K. Crawford. Critical questions for big data. *Information, Communication & Society*, 15(5):662–679, May 2012.

[4] A. Crabtree and R. Mortier. Human Data Interaction: Historical lessons from social studies and CSCW. In *Proc. European Conference on Computer Supported Cooperative Work (ECSCW)*, Oslo, Norway, Sept. 19–23 2015.

[5] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland. openpds: Protecting the privacy of metadata through safeanswers. *PloS one*, 9(7):e98790, 2014.

[6] C. Diaz, C. Troncoso, and A. Serjantov. On the impact of social network profiling on anonymity. In *Privacy Enhancing Technologies*, pages 44–62. Springer, 2008.

[7] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming*, pages 1–12. Springer Berlin / Heidelberg, Berlin, Germany, 2006.

[8] EU General Data Protection Regulation. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free

movement of such data. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2012:0011:FIN:en:PDF, 2012.

[9] T. Gazagnaire, A. Chaudhry, J. Crowcroft, A. Madhavapeddy, R. Mortier, D. Scott, D. Sheets, and G. Tsipenyuk. Irmin: a branch-consistent distributed library database. In *Proc. OCaml User and Developer Workshop at ICFP'14*, Sept. 5 2014.

[10] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis. Serving ads from localhost for performance, privacy, and profit. In *Proc. 8th ACM HotNets*, NY, USA, 2009.

[11] H. Haddadi, A. Chaudhry, J. Crowcroft, H. Howard, D. McAuley, A. Madhavapeddy, and R. Mortier. Personal data: Thinking inside the box. In *Proc. 5th Decennial ACM Aarhus Conference: Critical Alternatives*, Aarhus, Denmark, Aug. 17–21 2015.

[12] H. Haddadi, P. Hui, and I. Brown. Mobiad: private and scalable mobile advertising. In *Proc. 5th ACM MobiArch*, pages 33–38, New York, NY, USA, 2010. ACM.

[13] H. Haddadi, R. Mortier, S. Hand, I. Brown, E. Yoneki, D. McAuley, and J. Crowcroft. Privacy analytics. *SIGCOMM Comput. Commun. Rev.*, 42(2):94–98, Apr. 2012.

[14] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1223–1231. Curran Associates, Inc., 2013.

[15] Identity Theft Resource Center. ITRC Breach Statistics 2005—2015. http://www.idtheftcenter.org/images/breach/2005to2015multiyear.pdf, 2016.

[16] J. P. A. Ioannidis. Informed consent, big data, and the oxymoron of research that is not research. *American J. Bioethics*, 13(4):40–42, Mar. 2013.

[17] C. M. Johnson and T. W. A. Grandison. Compliance with data protection laws using hippocratic database active enforcement and auditing. *IBM Systems Journal*, 46(2):255 –264, 2007.

[18] A. Madhavapeddy, R. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft. Unikernels: Library operating systems for the cloud. In *Proc. ACM ASPLOS*, pages 461–472, 2013.

[19] R. Mortier, H. Haddadi, T. Henderson, D. McAuley, and J. Crowcroft. Human-data interaction: The human face of the data-driven society. Technical Report 2508051, SSRN, 2014.

[20] R. Mortier, T. Rodden, P. Tolmie, T. Lodge, R. Spencer, A. Crabtree, J. Sventek, and A. Koliousis. Homework: Putting interaction into the infrastructure. In *Proc. ACM UIST*, 2012.

[21] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proc. ACM Cloud Computing Security Workshop*, pages 113–124, 2011.

[22] D. Narayanan, A. Donnelly, R. Mortier, and A. Rowstron. Delay aware querying with Seaweed. *The VLDB Journal*, 17(2):315–331, Mar. 2008.

[23] E. Papadopoulou, A. Stobart, N. K. Taylor, and M. H. Williams. Enabling data subjects to remain data owners. In *Agent and Multi-Agent Systems: Technologies and Applications*, pages 239–248. Springer, 2015.

[24] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. CryptDB: processing queries on an encrypted database. *CACM*, 55(9), 2012.

[25] E. G. Rieffel, J. T. Biehl, W. van Melle, and A. J. Lee. Secured histories: computing group statistics on encrypted data while preserving individual privacy. *CoRR*, abs/1012.2152, 2010.

[26] Strategic Headquarters for the Promotion of an Advanced Information and Telecommunications Network Society. Policy outline of the institutional revision for utilization of personal data. http://japan.kantei.go.jp/policy/it/20140715_2.pdf, 2014.

[27] P. Tolmie, R. Mortier, T. Rodden, M. Önen, K. Elkhiyaou, and A. Friedman. D 4.1: Requirements, Ethics and Security Models for Privacy Preserving Data Management. Technical Report D4.1, EU FP7 User Centric Networking deliverable, Sept. 30 2014.

[28] US Consumer Privacy Bill of Rights. Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy. https://www.whitehouse.gov/sites/default/files/privacy-final.pdf, 2012.

[29] World Economic Forum. Rethinking personal data: A new lens for strengthening trust. http://www3.weforum.org/docs/WEF_RethinkingPersonalData_ANewLens_Report_2014.pdf, 2014.