Confidential Carbon Commuting

Exploring a privacy-sensitive architecture for incentivising 'greener' commuting.

Chris Elsmore Anil Madhavapeddy Ian Leslie Amir Chaudhry

Cambridge University Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FB, UK

first.last@cl.cam.ac.uk

Abstract

We discuss the problem of building a user-acceptable infrastructure for a large organisation that wishes to measure its employees' travel-to-work carbon footprint, based on the gathering of high resolution geolocation data on employees in a privacy-sensitive manner. This motivated the construction of a distributed system of *personal containers* in which individuals record fine-grained location information into a private data-store which they own, and from which they can trade portions of data to the organisation in return for specific benefits. This framework can be extended to gather a wide variety of personal data and facilitates the transformation of private information into a public good, with minimal and assessable loss of individual privacy.

This is currently a work in progress. We report on the hardware, software and social aspects of piloting this scheme on the University of Cambridge's experimental cloud service, as well as contrasting it to a traditional centralised model.

Categories and Subject Descriptors C.2.4 [*Distributed Systems*]: Distributed applications

General Terms Design, Measurement, Human Factors

Keywords Personal Containers; Privacy; Carbon Footprint; Mobile

1. Introduction

The University of Cambridge is currently engaged in reducing its carbon emissions by 34% [11]. One of the areas the University is exploring is the transport used by its 9,140 staff during their travel to and from work, which represented approximately 10% of the University's overall emissions for 2010 [10, 11]. To understand their employees' travel-to-work carbon footprint, the University conducts surveys[1], the results of which are used to plan future University and city infrastructure [9]. However, these results only reflect the small minority of the staff who respond, with variable accuracy due to manual self-reporting of routes taken to work.

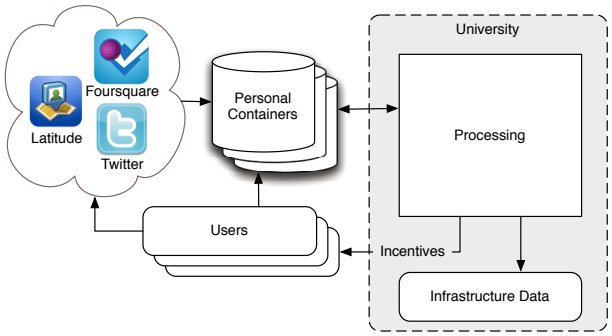

Figure 1. University can process data from the distributed personal containers of its staff. The containers can collate data from numerous sources, without divulging it to the University.

High resolution, continuously collected geolocation data from employees would significantly improve the accuracy of travel-to-work information available to the University. This would allow the collection of exact routes to work and would provide the ability to infer methods of transport, without employees needing to complete any surveys.

Since location data can be sensitive information, there are privacy considerations with this approach. Individuals may not be willing to openly share such sensitive data with their employer, and the employer may not wish to be responsible for holding and securing such information [2]. As the threat models regarding online privacy are constantly evolving, we are interested in exploring solutions which will be robust against future *long-term* threats, for example if the employer's privacy policy changed or if we wished to share information with other parties not yet envisaged.

[1] `http://www.admin.cam.ac.uk/offices/em/travel/`.

Eurosys '12, April 10th, 2012, Bern, Switzerland.

To this end, we propose a distributed system of *personal containers*[2], that can capture the benefits of access to high resolution location data whilst still respecting the privacy of employees (see Figure 1). These data-stores, owned by the individuals, can hold location data obtained automatically from devices such as mobile smartphones. We feel that using this infrastructure will also provide a platform for discussion of differential privacy; facilitating dialogue regarding distribution of trust, possible attacks and practical applications.

Such a system also reduces the need for a large, centrally managed data silo. Large silos allow for easier data processing but users typically have to trust the provider completely or not provide their data at all. In addition, once the data is added the user generally loses control over how it is exploited, where it is stored and even the ability to remove it.

Once implemented, this personal container framework can be extended to make use of a wide variety of data from employees with minimal loss of individual privacy, and be easily deployed by other organisations.

This is an ongoing project, and this paper covers issues we have encountered so far, and what remains to be completed. We discuss the potential issues of storing sensitive user data, the limitations of existing frameworks and how a personal container architecture addresses them (§2). The subsequent sections provide a more detailed description of how a personal container functions and how we set up the infrastructure within the University of Cambridge to collect and store user data (§3). This is followed by a brief summary of the upcoming work of developing specific applications, as well as using the data stored within personal containers to encourage reducing users' carbon footprints (§4).

In what follows, we will confine our discussion to the case of the employee providing specific portions of their data to the employer, recognising that this is likely to be less privacy-preserving than using a trusted aggregator or the user running a certified tool over their data.

2. Approaches to Data Collection

Collecting data from users' smartphones has serious privacy implications; due to its accuracy it can easily be used for nefarious activities such as tracking a user without their consent or knowledge. In the case of an employer, freely accessible location data (both historic and real-time) could be used prejudicially against an employee. A user must trust the data collection and storage providers before they allow their information to be stored, and be confident their data will only be accessed by parties they authorise and for the purpose(s) described.

To allow full control of access to their information, employees should be able to review a request for data by their employer before they allow it, and revoke that access at any time. They should also be able to impose limits as to which data each request can access, for example chronologically within a set time window, or geographically limited to a maximum deviation from a point or route. This enables the user to restrict access to likely times and routes to work, automatically excluding other journeys that occur outside of a usual commute time or path. A number of different architectures can be used to solve these issues, but there is an important trade-off between ease of use, cost and privacy to consider.

Mobile-only Tracking A single smartphone application could be used to record a user's commute to work by means of GPS or other geolocation data, and internally calculate which mode of transport was used and thus the carbon output. This ensures that data remains privately stored locally on the phone hardware and would generally only be exposed by losing the device.

However, resources on a typical smartphone are strictly limited for any inference calculation and this model prohibits any benefit from existing transportation data sources that the University may have available. It would only offer coarse benefits to the user, and the University would not benefit from having better travel-to-work data to augment their existing sources.

Hosted Website Tracking If the smartphone application were to send location data to a server hosted by an employer, this would permit them to benefit from the additional commute data, which allows the use of more significant resources to process the data and infer transportation. It also enables the employer to develop sophisticated incentives based on attributes such as users proximity to one another or related commutes, rather than the limited versions discussed above. However, if the website is unreachable (e.g. the smartphone loses signal), then more complexity is required on the mobile device to prevent gaps in the data from appearing. More significantly, the user has no option but to trust the server security, and no guarantee the data will not be used for other purposes they have not agreed to.

Hybrid Website/Mobile A bespoke website combined with such a smartphone application as described above suffers similar problems; users have no option but to trust the University to host their data, there exists no obligation to allow users access to their own information for other purposes, and the system becomes limited by the closed vertical stack, with the same data access risks as before.

Using an existing service such as Facebook as a gateway to the application brings additional benefits such as allowing users to log in with an existing username and allows powerful functionality such as using the social graph to identify friends with better commutes. However, excluding people who have not signed up to the underlying service limits the user base for the application and location information from the service, such as Facebook Places check-ins, does not provide enough data for transport inference. In addition,

[2] `http://perscon.net`

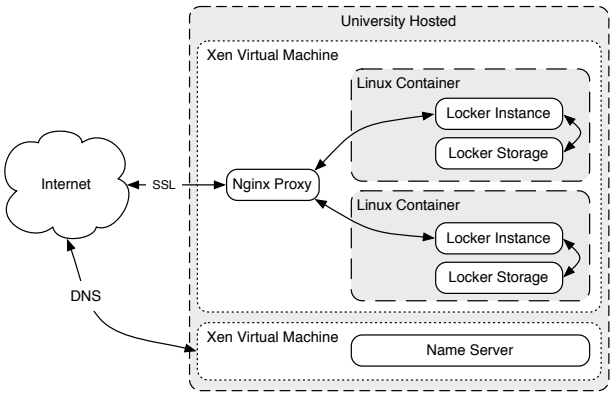

Figure 2. Layout of Personal Container deployment within the University.

the platform's evolving privacy policies are rarely met with universal acclaim [4, 5] and the data storage and access problems still exist.

Per-User Instances Our personal container architecture provides the benefits of the aforementioned systems without their disadvantages and concerns, by giving users more power and control over their data. It gives the user choice over where the container is installed and who by, and then gives the user direct control over which data in the container can be accessed. Users can store location data in their chosen container and allow the employer limited access to specific data in exchange for certain benefits. The employer can then provide applications that run inside the container, which openly request access to the user's data which can then be accepted or denied.

A personal container can be considered an individual's own data repository, and while it still requires trust in the hosting infrastructure, it gives the user the choice of which provider to trust, or to host it themselves.

3. Deploying Personal Containers

The deployment of personal containers has several aspects that make it more complex than a conventional centralised application: (1) distributed identity management, (2) economical access to computation resources that are sufficiently isolated from each other, (3) routing data gathered from end-user devices directly to their personal container and (4) distribution of costs related to infrastructure and maintenance.

To prototype the system, we based the core of our personal containers on code from The Locker Project[3], an open-source effort in which users can store information and create applications that run inside *lockers*. For the purposes of the following sections, we consider the terms *personal container* and *locker* to be interchangeable.

[3] `http://lockerproject.org`

Whilst the open-source locker codebase is suitable for a single individual deployment, we had to extend it with support for hosting thousands of users, while maintaining a reasonably low cost-per-user (§3.1). For our initial pilot, all user lockers are hosted on University infrastructure, but the architecture is explicitly designed to support hosting lockers elsewhere (§3.2). A variety of location sources are supported using popular cloud services, as well as a local data gathering app on the mobile phone only (§3.3). One development that greatly aided our prototype was that we became early users of a University "private cloud" based on the Xen Cloud Platform[4], which provided a cost-effective way to prototype and subsequently scale our service (§3.4).

3.1 Identity Management

When using a centralised web service, identities are usually handled by self-registered accounts. In our deployment, we construct and maintain a locker for each member of staff to store their private data. Therefore we integrate with existing University-wide systems, taking advantage of unique usernames that are used within the University and never reused, even if that person leaves the organisation. We map this id onto DNS to assign all staff an individual locker via a unique hostname of the form `<id>.locker.cam.ac.uk`. A similar system could be setup within other organisations using pre-existing usernames.

The locker service is accessed solely through SSL-encrypted TCP connections. The SSL certificates are unique per user, and currently issued from a self-signed authority. In production, each hostname's key will be derived from a secret associated with that user in the University identity database. This means that staff can gain secure access to their locker service without remembering any additional passwords. This is similar to how staff connect to the widely deployed Eduroam[5] wireless service, via one-time tokens derived from their identity.

The main challenge we faced at this stage was one of administrative processes to register the sub-domain and operate custom name servers — since DNS is a critical network service, most IT organisations are reluctant to permit experimental services to run on it. In the end, our deployment only required a single delegation from the production network (for the `NS` record for the `locker` subdomain), and the use of the private cloud service (§3.2) made the remainder of the infrastructure deployment much more straightforward.

This DNS indirection is simpler and shorter for end-users than encoding the username in a URL path, and the specific network location of an individual's locker can now be controlled simply by a nameserver update to point to a new location. This architecture will not preclude users from implementing their own locker elsewhere if they wish.

[4] `http://xen.org/products/cloudxen.html`

[5] `http://www.eduroam.org`

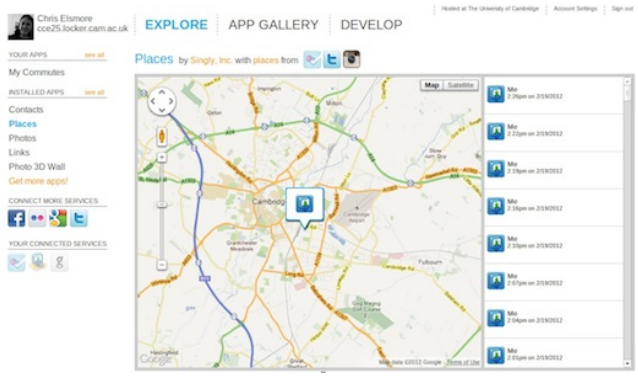

Figure 3. A prototype locker running a mapping application.

3.2 Anatomy of a 'Locker'

Using the infrastructure described above, a user can now access their own individual and isolated lockers hosted by the University or elsewhere. Figure 2 shows the overall architecture of how the locker deployment is structured within the University, which we now explain in more detail.

3.2.1 Locker Structure

Lockers are implementations of a personal container, equally suited for hosting on a user's home computer, a cloud based platform or by an existing organisation. The codebase is written in Javascript using the Node.js[6] platform for the backend, HTML and Javascript for the web-based user interface, and uses a MongoDB[7] database for data storage.

Each locker features 'connectors' that are responsible for connecting to outside services such as Facebook or Twitter, and pulling in new information. This new information is deduplicated and then stored in the database by the locker core service, which also controls how often external services are polled for new data. Lockers also provide an internal API to the stored data, used by installable locker web applications which then present this information to the user. Figure 3 shows the locker web interface running a simple mapping application.

3.2.2 Deployment

Although the lockers are currently hosted on University infrastructure, this is simply a means of bootstrapping the system for the current pilot project. The intention is that individuals can take full ownership of their data-stores and move off the University hardware whenever they wish.

Our deployment of personal containers involves numerous instances of the locker platform. It was critical to ensure these multiple instances did not interfere with each other, thus we have deployed multiple layers of virtualisation to improve the isolation between them, as depicted in Figure 2. Firstly, groups of users are assigned a virtual machine hosted on the Xen Cloud Platform, provided by the University Computing Service[8].This VM is further subdivided into smaller containers using LXC[9] Linux Containers for each individual user. This allows groups of users and their data to be manipulated and backed up at the VM level, and ensures individual privacy in a cost-effective way.

Access to each user's locker is only available through an Nginx[10] proxy server, which controls authentication to each locker and prevents unauthorised access. Name-based virtual hosting for SSL is not fully supported by all browsers, therefore we use this proxy to route users to their individual locker based on hostname, and to secure communications between the proxy and client using SSL as discussed previously.

3.3 Location Data Gathering

To gather high-quality location data, we are developing a mobile application that interfaces directly with individual lockers, rather than go via a third-party social networking service (§4.3). As well as this direct data, our deployment of personal containers can already gather location information from a number of social services that users may be registered with. These are implemented as Locker Connectors, and currently comprise of Facebook and Foursquare check-ins, geotagged information from Twitter and Flickr and also updates from Google Latitude. These sources are all available to any user who wishes to connect them to their locker via their web interface (see Figure 3).

3.4 Distribution of infrastructure costs

The pilot programme involves the creation of both containers and mechanisms for the University to query the data, thus the initial costs are borne by the current project. However, long-term sustainability depends upon ongoing costs being distributed in a manner that reflects usage. Considered this way, the costs can be divided into two components.

Firstly, there are costs associated with maintaining hosted personal containers (see Figure 2). Although these are 'owned' by the individual users, costs could be subsidised by employees' departments, in the same way that overheads such as pension-contribution plans and healthcare benefits are. A significant benefit of using a private cloud infrastructure for the hosting is that we can track resource usage at a per-user level.

Secondly, there are the costs of running the central data processing systems (shaded area in Figure 1). These systems process data that employees have made available. As this is a centralised utility, it can be paid for and regarded in the same manner as the existing travel-to-work survey.

Finally, there may also be an opportunity for cost-savings. Actual travel-to-work information is likely to be of higher

[6] http://nodejs.org/

[7] http://www.mongodb.org/

[8] http://www.ucs.cam.ac.uk/

[9] http://lxc.sourceforge.net/

[10] http://nginx.org/

value than data gained through the existing manual travel-to-work surveys. Therefore, as more employees "respond" via smartphones and personal containers, resources can be re-allocated from running surveys to other areas.

4. Design Principles and Ongoing Work

The infrastructure described so far is relatively low-level, but serves as a useful basis for building a more distributed, privacy-friendly data processing platform.

4.1 Trusted Aggregators and Differential Privacy

The results of querying actual location history will yield significantly better results than those gained via survey questions. Consider the following query:

"What time do you arrive at work?"

If asked via a survey, the likely response might be 9am, as most employees are expected to be working from then. However, the same query run over users' *actual* location data might reveal that a proportion of employees are systematically late, arriving at 9:30am. This is an example where accurate information could better inform infrastructure needs but divulging such information individually could have punitive consequences for employees.

This leads to a privacy issue as running further queries against a user's route to work, coupled with their arrival time, could start to de-anonymise data and possibly identify these users [8]. As the granularity of data available for processing increases, it exposes interesting questions regarding the ability to identify individual users to the University or other third parties.

Such queries can still be performed while reducing privacy loss or exposure, with the addition of a randomised response system [3, 12]. Such an addition would allow a *trusted aggregator* to perform queries across a number of lockers. This aggregator is assured to not reveal identifying information by the addition of random sampling and noise to responses to hide individual answers but retain the overall view as the noise is added to all respondents. Another alternative technique that a trusted aggregator could employ is "differential privacy", to guarantee anonymity limits for external queries by examining the full dataset [7].

```
for locker in lockerList:
    if random():
        result = locker.query()
        privResults += result * noise.random()

return privResults / privResults.length()
```

Across many personal containers, a trusted aggregator can be used to query the most popular time locker users arrive at work for example - the addition of noise does not affect the distribution of answers. The pseudocode snippet above shows an oversimplified version of how this might be undertaken. This represents a differential privacy architecture such as described by R. Chen *et al* [1], and represents a significant avenue for investigation and will be discussed in future work.

4.2 Locker Data Access & Applications

The existing locker codebase features very little data access control; currently every application is allowed access to all user data inside the locker while a list of icons beside each application is used to advertise to the user which data sources will be accessed. These are currently set in a configuration file, and there are no guarantees an application will be well behaved and not access other data it has not advertised.

Thus, applications we produce to educate and inform the user of their travel-to-work footprint must implement their own data checking and confirmation layer, to allow the user the control over their data outlined at the beginning of this paper. The locker itself exposes a query API to request data, and we are experimenting with creating middleware that can take requests from this API, confirm them with the user and then act on them if the user consents [6].

This allows an application to be constructed that will utilise both the higher resolution data gathered from the smartphone application, as well as using the confirmation layer (as described above) to verify the app is allowed to access the users location data, to calculate the carbon information and convey this to the user.

Using this application in combination with smartphone data, the University will receive significantly more informative data for their future planning projects than can currently be obtained by the existing 'travel-to-work' survey.

4.3 Smartphone Application

In future work we plan to explore the development of an application for smartphones that is capable of logging high resolution, highly accurate geolocation data.

This app would be specifically designed to only capture a user's commute, allowing the restriction of data recording by setting either manually or automatically a time or geographic window. Movements that occur outside of this window would not be recorded in order to preserve privacy, with the added benefit of a reduction of battery usage compared to a full-time location logger.

5. Summary

We have discussed our ongoing work to design and prototype a privacy-sensitive architecture within the University of Cambridge. We believe the personal container architecture is an important improvement to current traditional vertically integrated services, allowing the user a great deal more visibility and control over their own data. Using this we can create a powerful framework for incentivising the reduction of users' travel-to-work carbon footprint. Underpinning this is a need to be more future-proof to future privacy needs by not constructing large data silos, but instead computing "closer"

to the data and letting individual users reveal it selectively to their employer.

5.1 Future Possibilities

As mentioned previously, creating this infrastructure provides a platform for future discussion of differential privacy, distribution of trust, possible attacks and practical survey applications. Using this infrastructure, an organisation could offer a range of schemes to its employees beyond commuting information, in a privacy-sensitive manner and with the explicit permission of the users. Further schemes can be run, building upon the travel-to-work scheme as well as using alternative data sources and encouraging different behaviours.

For example, we could extend the commuting scheme by calculating how much money a user spends using a car, train or bus during their commute, and automatically indicate the payback time of using the Cycle to Work Scheme[11] to buy a bicycle, including when the user breaks even.

Locker users could also gain longer term benefits beyond the University's travel-to-work projects. Users could store high resolution gas, water and electricity consumption information from their homes, and locker applications could inform them of their overall consumption. That data could then be used to offer replacement recommendations for old inefficient appliances, or automatically recommend energy tariffs that are cheaper based on recorded usage.

As discussed above, the data involved in these services is directly controlled by the user; each scheme will stipulate exactly which data it needs and why, the user being free to revoke access at any time. In addition, the co-location of both the data and application in an open framework gives users greater control, as well as lowering the barriers for creating their own applications for use with their own data. We believe this control coupled with the flexibility of hosting personal containers on cloud platforms that users trust, be they the University's, within users' own homes or elsewhere, will result in the majority of users being happy to store their personal data in such a container as the potential benefits are significant.

References

[1] R. Chen, A. Reznichenko, and P. Francis. Towards statistical queries over distributed private user data. In *proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation*. USENIX, 2012.

[2] J. Crowcroft, A. Madhavapeddy, M. Schwarzkopf, T. Hong, and R. Mortier. Unclouded vision. In *proceedings of the 12th International Conference on Distributed Computing and Networking*, ICDCN'11, pages 29–40. Springer-Verlag, 2011. ISBN 3-642-17678-X, 978-3-642-17678-4.

[3] W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 505–510. ACM, 2003. ISBN 1-58113-737-0. doi: 10.1145/956750.956810.

[4] B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *proceedings of the first workshop on Online Social Networks*, WOSN '08, pages 37–42. ACM, 2008. ISBN 978-1-60558-182-8. doi: 10.1145/1397735.1397744.

[5] B. Krishnamurthy and C. E. Wills. On the leakage of personally identifiable information via online social networks. *SIGCOMM Comput. Commun. Rev.*, 40(1):112–117, Jan. 2010. ISSN 0146-4833. doi: 10.1145/1672308.1672328.

[6] D. McAuley, R. Mortier, and J. Goulding. The dataware manifesto. In *COMSNETS'11*, pages 1–6, 2011.

[7] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *proceedings of the 35th SIGMOD international conference on Management of Data*, SIGMOD '09, pages 19–30. ACM, 2009. ISBN 978-1-60558-551-2. doi: 10.1145/1559845.1559850.

[8] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125. IEEE Computer Society, 2008. ISBN 978-0-7695-3168-7. doi: 10.1109/SP.2008.33.

[9] U. of Cambridge. Travel plan. URL `http://www.admin.cam.ac.uk/offices/em/travel/plan/Travel_Plan_2011.pdf`.

[10] U. of Cambridge. Travel for work survey report, 2010. URL `http://www.admin.cam.ac.uk/offices/em/travel/plan/TfW_2010_Survey_Report_University_of_Cambridge.pdf`.

[11] U. of Cambridge. Carbon management plan 2010–2020, Sept. 2010. URL `http://www.admin.cam.ac.uk/offices/em/sustainability/environment/climate/cmp.pdf`.

[12] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft. Spotme if you can: Randomized responses for location obfuscation on mobile phones. In *Proceedings of the 31st International Conference on Distributed Computing Systems*, ICDCS '11, pages 363–372. IEEE Computer Society, 2011. ISBN 978-0-7695-4364-2. doi: 10.1109/ICDCS.2011.79.

[11] `http://www.admin.cam.ac.uk/offices/hr/staff/benefits/cycle/`