Confidential Carbon Commuting
Exploring a privacy-preserving 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 an acceptable infrastructure for a large organisation that wishes to measure its employees’ travel-to-work carbon footprint, which requires the gathering of high resolution geolocation data on employees in a privacy-preserving 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. The knowledge gained can then be used to create schemes to encourage behavioural change, with the aim of optimising overall energy use.

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

1. Introduction
The University of Cambridge is currently engaged in reducing its carbon emissions by 34% [1]. One of the areas the University is exploring is the transport used by its 9,110 staff [2] during their travel to and from work, which represented approximately 10% of the University’s overall emissions for 2010 [3]. 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 needs [4]. 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.

High resolution, continuously collected geolocation data from employees would significantly augment 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 is highly sensitive information, there are important privacy concerns with this approach. Individuals may not be willing to openly share such sensitive data with their employer or in general, and the employer may not wish to be responsible for holding and securing such information [5]. In order to capture the benefits of access to high resolution location data whilst still respecting the privacy of employees, we propose a distributed system of personal containers. These data-stores, owned by the individuals, can hold location data obtained automatically from devices such as mobile smartphones. Individuals can then trade specific portions of that data to the University, in return for specific incentives. In effect, the University places a value on this personal data (on the basis of long-term infrastructure savings), and passes on some of this to staff in return for analysing their private data, as shown in Figure 1.

This is an ongoing project, and this paper will cover interesting issues we have encountered so far, and what remains to be completed. We discuss the potential issues of storing

\(^1\)http://www.admin.cam.ac.uk/offices/em/travel/.
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 behavioural change towards reducing carbon emissions (§4).

2. Privacy-Preserving Data Collection

The data collected 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 the University 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. A number of different architectures can be used to solve these issues, but there is a trade-off between ease of use, cost and privacy to consider.

Mobile-only Tracking For example, 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 the University, 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. Despite those issues, this model permits the University 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 University to develop sophisticated incentives based on attributes such as user proximity to one another or related commutes, rather than the limited versions discussed above. However, if the website is unreachable to the smartphone (e.g. during train travel), then more complexity is required on the mobile to prevent holes in the data from appearing.

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. Furthermore, powerful functionality such as using the social graph to identify friends with better commutes is possible. However, excluding people who have not signed up to the underlying service limits the user base for the application, and the sole use of user location information from the underlying service, such as Facebook Places check-ins, does not provide enough data for transport inference. Services such as Facebook also have a tarnished record for user privacy [6–9] and the data storage and access problems still exist.

Per-User Instances Our personal container architecture provides the benefits of the aforementioned systems without these concerns, by giving the users more power and control over their data. A personal container can be considered an individual’s own data repository, possibly hosted by a trusted third party, which the University must explicitly be granted access to.

This architecture 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 University limited access to specific data in exchange for certain benefits. The University can provide applications that run inside the container, which openly request access to the user’s data. These requests can be allowed or denied by users. An application can perform data processing using external data sources if necessary, and provides the user with the benefits of being included in any incentive schemes the University decides to run, while allowing users to retain access and control over their personal data.

3. Deploying Personal Containers

The deployment of University-wide 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 infrastruc-
ture and maintenance.

To prototype a system within the University, we based the
core of our personal containers on code from The Locker
Project\(^2\), an open-source effort in which users can store
information and create applications inside locker databases.
For the purposes of the following sections, we consider the
terms personal container and locker to be interchangeable.

Whilst the open-source lockers are 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 by the University, but the architecture is explic-
itly 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 on the mo-
bile phone only (§3.3). One development that greatly aided
our prototype was that we became early users of a Univer-
sity “private cloud” based on the Xen Cloud Platform, 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, and thus must integrate with exist-
ing University-wide systems. Every member of staff has a
unique username within the University that is never reused,
even if that person leaves the organisation. Therefore, we use
this id with the DNS to assign each member of staff an in-
dividual locker by means of a unique hostname of the form
<id>.locker.cam.ac.uk.

The locker service is accessed solely through SSL-
encrypted TCP connections. The SSL certificates are unique
per user, and currently signed from a self-signed authority.
In production, each hostname’s key will be derived from
a secret associated with that user in the University iden-
tity 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\(^3\) wireless service, via one-time tokens
derived from their identity.

The main challenge we faced at this stage was one of ad-
ministrative processes to register the sub-domain and oper-
ate custom name servers — since DNS is a critical network
service, most IT organisations are reluctant to permit exper-
imental services to run on it. In the end, our deployment
only required a single delegation from the production net-
work (for the NS record for the locker subdomain), and the

\(\text{\footnotesize 2http://lockerproject.org}
\)

\(\text{\footnotesize 3http://www.eduroam.org}
\)

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. Other
services that use DNS like this for service discovery include
XMPP messaging.

3.2 Anatomy of a ‘Locker’

Using the infrastructure described above, a user can now ac-
cess their own individual and isolated lockers hosted by the
University or elsewhere. Figure 2 shows the overall archi-
tecture 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\(^4\) platform for the
backend, HTML and Javascript for the web-based user in-
terface, and uses a MongoDB\(^5\) database for data storage.

Each locker features ‘connectors’ that are responsible for
connecting to outside services such as Facebook or Twit-
ter, 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 applica-
tions which then present it to the user. Figure 3 shows the
locker web interface running a simple mapping application.

\(\text{\footnotesize 4http://nodejs.org/}
\)

\(\text{\footnotesize 5http://www.mongodb.org/}
\)

\[\text{\footnotesize Figure 2. Layout of Personal Container deployment within the University.} \]
The Locker platform is made up from 4 main components:

- **Connectors** are responsible for connecting to and downloading data from services such as Flickr or Facebook. They handle authenticating to the requested external service, checking for new data and transforming this data to the Locker Core.

- **Core** is the central Locker process that fires events such as instructing connectors to check and update data, as well as storing this data in the datastore. It also provides a query API to the stored datastore, which applications can use to retrieve and display data to the user.

- **Applications** interact with the Locker Core, requesting a subset of data from the locker and rendering it to the user.

- **Datastore** for deduplicated database storage.

### 3.2.2 Deployment

Our deployment of personal containers within the University 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. Firstly, groups of users are assigned a virtual machine hosted on the Xen Cloud Platform[^6], provided by the University Computing Service[^7]. This VM is further subdivided into smaller containers using LXC[^8] 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[^9] proxy, 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

Currently our deployment of personal containers can gather location information from a number of social services that users may already be registered with. These are implemented as Locker Connectors, and currently comprise of Facebook and Foursquare check-ins, geo-tagged information from Twitter and Flicker and also updates from Google Latitude.

These sources are all available to any user who wishes to use them, via a web interface (recall that every user has a unique `<id>.locker.cam.ac.uk` domain of their own). Some early adopters of the service have already commented that they are uncomfortable with giving their location information away to a third-party cloud service. To solve this, and also gather higher-quality data, we are developing a mobile application that interfaces directly with individual lockers (§4.3).

### 3.4 Distribution of infrastructure costs

So far, we have not discussed the cost implications of deploying and maintaining the personal container infrastructure. The pilot programme involves the creation of both containers and the mechanisms to trade data with the University, thus the initial costs are borne by the current project. However, the long-term sustainability of the infrastructure depends on the ongoing costs being distributed in a manner that reflects how the data is used. When considered this way, the costs can be divided into two components.

Firstly, there are costs associated with maintaining the hosted personal containers (see Figure 2). Since these are ‘owned’ by the individual users, the costs of maintaining them could be borne by the departments where the users are employed. In this way, the costs would simply become part of the ‘overhead’ that departments incur for each employee. Other examples of overhead costs are pension-contribution plans and healthcare benefits. In fact, secure hosting of personal containers could be considered an employee benefit. 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 maintaining and running the central University data processing systems (the shaded area in Figure 1). These are the systems which receive and process information that employees have made available, and which the University can use to understand the overall travel-to-work carbon footprint. 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 the possibility of cost-savings in other areas. Actual travel-to-work information, released by employees, is likely to be of substantially higher quality.

[^6]: http://xen.org/products/cloudxen.html
[^7]: http://www.ucs.cam.ac.uk/
[^8]: http://lxc.sourceforge.net/
[^9]: http://nginx.org/
than can be gained from the existing travel-to-work survey. Therefore, as more employees use smartphones and personal containers, resources could be re-allocated from the surveys to more effective methods of interacting with employees.

4. Next Steps

There are three main tasks to address following on from our position currently. Once we have completed these key tasks, we will be in a better position to fully realize and discuss the benefits that using a distributed system of personal containers provides. As this is an ongoing project, we welcome comments and constructive criticism.

4.1 Locker Application

We are in the process of designing an application to run within the personal containers of participating users that will access the stored location information, and infer from it the route and mode of transport of the user’s commute. This application will thus be able to calculate the annual carbon emissions generated and relay this information to both the University and the user, to increase awareness of their own carbon footprint.

The University can then use this application to facilitate behaviour change by both advising users of routes that are more efficient or cheaper, as well as running schemes to further encourage fewer carbon emissions, such as ride shares or cycle to work projects. The University will also receive significantly more informative data for their future planning projects than can currently be obtained by the existing ‘travel-to-work’ survey.

The use of a distributed architecture means users who are unwilling to subscribe to a service hosted by the University are still provided with access to the same offers available to owners of University-hosted containers.

4.2 Data Access Confirmation Layer

The existing locker codebase currently features very little data access control; every application has access to all of the data inside the locker. Presently, 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 is no guarantee an application will be well behaved and not access other data it has not advertised. A more rigorous solution is in development for integration into the main locker-project codebase, but it has yet to be defined.

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. In addition, since the locker is open source, source code from each application a user can install is available, should the user wish to check exactly what the application does before using it.

4.3 Smartphone Application

We also plan to explore the development of an application for smartphones that is capable of logging high resolution, highly accurate geolocation data. An initial overview of the data available from a typical locker reveals Google Latitude data to be the probable source of the highest quality data, however the resolution may not be sufficient for inference of shorter intra-city transport methods. In addition, the official Latitude client is currently unable to record a user’s location without a working network connection.

Existing phone logging applications exist, such as the Android application Device Analyzer\(^\text{10}\), which can log a significant amount of information from the user’s phone. This existing project could be augmented to suit our needs, or a new dedicated application could be developed. Independent of the basis of the application, it would be specifically designed to only record 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.

Whilst using an external tracking service such as Latitude can have privacy issues, this is less of an immediate issue than a user’s employer having access to the same data, for reasons mentioned previously. As the quality of data available for processing increases, it exposes interesting questions regarding the ability to identify individual users to the University or other third parties. The need for a randomised response or differential privacy system such as the work by R. Chen et al\(^\text{10}\) being included in our deployment is another avenue for investigation and will be discussed in future work.

5. Summary

We have discussed our ongoing work to design and prototype a privacy-preserving architecture within the University of Cambridge, based on an implementation of the personal container paradigm called locker. 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.

5.1 Future Possibilities

As shown in Figure 4, using this infrastructure an organisation could offer a range of schemes to its employees beyond commuting information, in a privacy-preserving manner and

\(^{10}\)http://deviceanalyzer.cl.cam.ac.uk
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\(^\text{11}\) to buy a bicycle through salary sacrifice, including when the user breaks even and becomes better off.

The University could also offer financial benefits by storing high resolution gas, water and electricity consumption information from a user’s home, to inform them of their consumption. This data could then be used to offer replacement recommendations for old inefficient appliances, or recommend energy tariffs that are cheaper based on recorded usage, automatically.

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 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 user’s 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.

\(^{11}\text{http://www.admin.cam.ac.uk/offices/hr/staff/benefits/cycle/}\)

References


