

# Dataccount System Overview

## Problem statement

Many companies maintain databases of personal information about individuals, which they use for their own purposes:

- Web sites like Facebook, Amazon, and Google monitor your online activities, then sell the information to advertisers for their own profit, with no compensation to you.
- Roomba's robot vacuums make a map of the interior of your home and send it back to the company. Roomba uses it to improve their robot algorithms, again with no compensation.
- Amazon keeps a permanent database of your personal home address and credit card information, which they can use as they see fit.
- Doctors keep your sensitive medical history in their own databases. You have no control over it.

So, the first problem is that information about yourself is stored by many companies, who generally promise to keep it safe, but you have no concrete way of verifying it. The companies may accidentally expose information which is private or embarrassing, or which allows other people to impersonate you. They can make money from your information without compensating you. You just have to trust them.

The second problem is that by keeping individuals' information, companies are exposed scandal, lawsuits, etc. if they lose the information, for example to hackers.

## Proposed solution

This document proposes a combination of computer systems and laws that would:

- A. Allow companies to access personal information under the control of the individual.
- B. Allow individuals to track accesses to their own personal information.
- C. Prevent companies from storing personal information.

## A. Allowing access under individual control — Example

When I order something on Amazon today, they show me a screen that contains my address and credit card, as shown on the left. This shows how Amazon maintains my personal information in their private database.

The image on the right shows the same screen after Amazon has upgraded to use this system. All personal information is gone. Instead there is only a web address for a “dataaccount” (“data” + “account”). This shows how Amazon doesn’t keep my personal information. They get it from my dataaccount, which is at “https://merrittmarcthompson-dataaccount”:

The image displays two side-by-side screenshots of an Amazon checkout screen, illustrating a transition from a traditional personal information-based system to a dataaccount-based system.

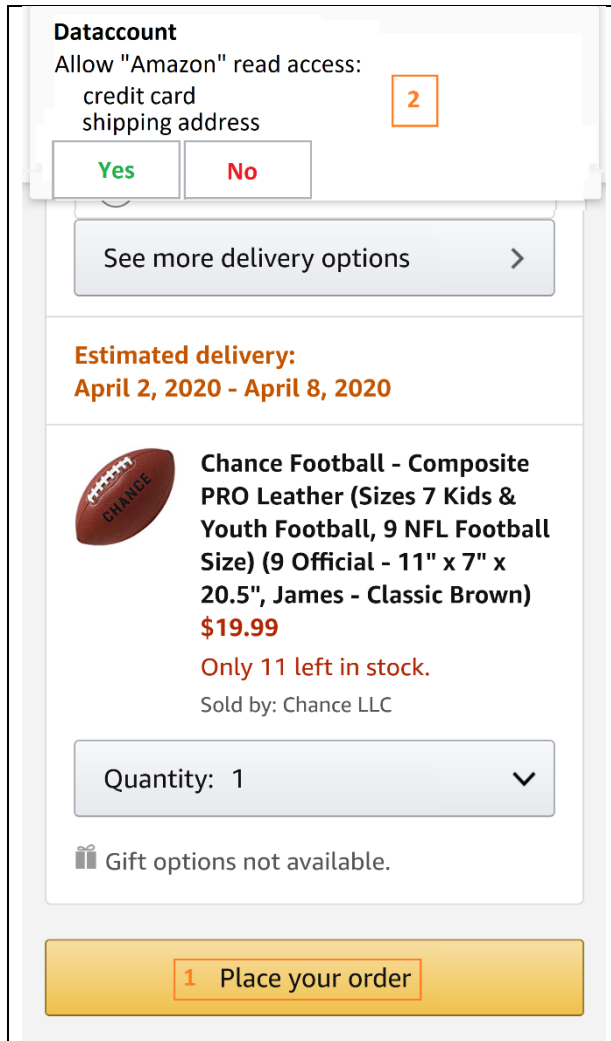
**Left Screenshot (Traditional System):**

- Estimated delivery:** April 2, 2020 - April 8, 2020
- Items:** \$19.99
- Shipping & handling:** \$0.00
- Total before tax:** \$19.99
- Estimated tax to be collected:\*** \$0.00
- Order total:** \$19.99
- Shipping address:** Merritt Thompson, 2250 Fernwood Cir, Lake Oswego, OR 97... (highlighted with an orange box)
- Add delivery instructions (optional)** >
- Payment information:**
  - Payment method:** Visa ending in 1126 (highlighted with an orange box)
  - Billing address:** Same as shipping address >
  - Get a \$100 Amazon.com Gift Card** > (upon approval for the Amazon.com Store Card)

**Right Screenshot (Dataaccount System):**

- Estimated delivery:** April 2, 2020 - April 8, 2020
- Items:** \$19.99
- Shipping & handling:** \$0.00
- Total before tax:** \$19.99
- Estimated tax to be collected:\*** \$0.00
- Order total:** \$19.99
- Dataaccount:** https://merrittmarcthompson-dataaccount For "credit card" and "shipping address" (highlighted with an orange box)
- Add delivery instructions (optional)** >
- Payment information:**
  - Get a \$100 Amazon.com Gift Card** > (upon approval for the Amazon.com Store Card)

When I click the Amazon "Place your order" button (1 below), Amazon tries to access my dataaccount. That causes a push notification from the Dataaccount software on my phone (2), which lets me approve the access. This allows Amazon to bill my credit card and print a shipping label without permanently storing my personal data:



## B. Tracking access to personal information

When a company accesses my data, my dataaccount software logs the access and its purpose:

Access date/time	Accessor	Data Item	Access	Allowed?	Purpose
4/7/2025 19:03:27	www.amazon.com	credit card	Read	Yes	Purchase \$19.99 football
4/7/2025 19:03:27	www.amazon.com	shipping address	Read	Yes	Print delivery label

## C. Restricting storage of personal information

Thus far, this system gives external entities a way to access necessary information without storing it, but it doesn't stop them from doing so anyway. I.e. they can still access the information from the dataaccount, then save it privately for their own use or profit.

This would be prevented by new laws that would criminalize storage of other people's personal data for purposes other than the stated ones.

In the example above, if Amazon got your shipping address, then used it for any purpose other than printing a delivery label (for example, if they sold it to a mass-mailing company), they would have committed a crime.

This law would criminalize what is now an extremely common practice. You might think that companies would oppose this restriction, but it is more likely that they would ultimately approve of it, because as long as they follow the rules, they are completely absolved of liability for individuals' data losses.

## Compensation

In this scheme, your Roomba vacuum could make a floor plan of your house, then send it directly to your dataaccount. Roomba could then buy your floor plan from you. I'm not sure if it would be that lucrative, but Roomba might give customers perks of some kind for it. You could also sell it to researchers other than Roomba, since it is your data, not Roomba's.

Another example like this would be selling medical data to researchers. If you have a rare or interesting condition, you might get a reasonable amount of money for your medical records.

## Implementation overview

### Dataaccount location

A dataaccount can go on any server that is on the web, so long it supports the dataaccount web API. Thus, you could have your own server locked in a closet in your house. Or you could contract with a "data bank" company that provides encrypted dataaccount services for many individuals.

### Data format

Personal data would be stored in the form of key/value pairs, for example:

"Home address"="Jane Smith, 123 Shady Lane, Tulsa, Oklahoma, 74101"

## Nondisclosure of data in keys

Imagine that you want to send a diamond ring to your mistress Sheila's home address. Let's say you already have a data key called "My mistress Sheila's home address". So, you might fill out a field like this on the Amazon web site:

Shipping Address Data Key:     `My mistress Sheila's home address`

This would be less than ideal. To prevent this, you could create an *alias* in your dataaccount:

`"Amazon shipping address"="@My mistress Sheila's home address"`

This means, "Whenever someone asks for 'Amazon shipping address', give them 'My mistress Sheila's home address' instead".

Amazon would default their form to the following, so you probably wouldn't have to enter anything:

Shipping Address Data Key:     `Amazon shipping address`

## APIs

There would be two APIs:

- One for individuals ("owners") to manage their own personal information. This includes functions that support the on-the-fly push notifications for Allow and Deny illustrated above.
- One for companies ("accessors") to access personal information of others. This is just one function to access owner data, providing what access is requested

## Owner authentication

Owners can be authenticated for the owner API using HTTP Basic authentication or OAuth. Encrypted HTTPS transport must be used for HTTP Basic to work, as the password is merely obfuscated.

## Accessor authentication

The most important thing for the accessor API is to authenticate the accessor. For example, if someone tried to access your dataaccount, said they were "Amazon", and the dataaccount simply believed it without authentication, anyone could impersonate Amazon and get your information.

Transport Layer Security (TLS), the basis of the HTTPS communication protocol, normally requires the server to prove who they are to the client before any data can pass between them. The server does this by using a secret private key to encrypt some data. If the client can successfully decrypt the data using the public key, it proves that the server is who they claim to be.

But that process is the opposite of what this system requires. In this system the server (the dataaccount) must obtain proof that the client (the accessor, ex. Amazon) is really who they claim to be, not the other way around.

Fortunately, there is a "two-way" variant of TLS which provides exactly this function. It both verifies that the server is who they claim to be to the client (as normal) and the client is who they claim to be to the server (needed for this system).

## **Push notifications**

When a company contacts your dataaccount web API to access data, your server must immediately notify your mobile device for approval. It's possible to set up a "service worker" via JavaScript, which always runs and waits for such notifications.