Master Thesis: Open Specification of a user-controlled Web Service for Personal Data

G. Jahn

November 7, 2016

Abstract

Data is the currency of tomorrow. Organizations, whether in the private or public sector, gathering enormous amounts of personal (big) data. This data is harvested and incorporated by these third parties, but were created by individuals and therefore should belong to them. People depending on all their data. Their identity as well as their personality are defined by their personal data. Meanwhile data silo operators are hammering onto these haystacks eager trying to find any kind of correlations worth interpreting, thereby almost inevitably discriminating the rightful owners. To reduce the possibility of discrimination only the least amount of data that is required should be handed over to the third party. Thus the individual has to be in charge of the whole process. A personal data service will empower its user to regain full control over her data and facilitates detailed information on every data flow. To be able to trust such a tool, the user should be able to look inside. Therefore a personal data service has to be open source and developed transparently, which then also allows to self-host it.

Contents

1

Introduction

1.1 Motivation

Nowadays it is rare to find someone who doesn't collects data about some kind of things, particularly humans are the targets of choice for the *Big Data Movement*. But since humans are all individuals, which means we are all distinct from each other - more or less. Some of us are sharing a bit more similarities, but most are much less similar to each other. Since these few similarities are widely shared, they should be less important, because it's the nature of inflationary occurrence. But the opposite happens to be the case.

In addition humans intend to create more and more data every da - proactively by writing a tweet and passively by appending their current location to that tweet when submitting it. So there are huge amount of data to collect, aggregate and analyze.

So it happens

> In the context of civil rights law, discrimination refers to unfair or unequal treatment of people based on membership to a category or a minority, without regard to individual merit. *[1]*

- It is a matter of individual perception what's good or bad, what's armful or what's an advantage. Collecting, aggregating and analyzing data is foremostly just science and technology. It is up to the humans how such tools are getting used and what purposes they are serving. Hence it should be decided by the data creators, what's happening to their data and what parts of them are used.

- Additionally

- Not only when interacting directly with each other discrimination of human beings seems to be still a serious issue in our society, but also when humans leveraging computers and algorithms to uncover former unnoticed information to support them in decision making - for example when granting credits, hiring employees, investigating crimes or renting flats. Approving or denying is all happening based on data about the affected individuals. That's all in fact discrimination, but on a much larger scale and rate with less effort - and almost parenthetically. This is know as the discriminate humans related to certain data, by interpreting results of deep learning and using neuronal networking and trying to extract information out of data bulks

- It is all about understanding the human being and why she behaves as she does. The challenge is not only to computing certain motives but rather concluding to the right ones. When analyzing computed results with the corresponding data models and trying to conclude, it is important to keep in mind, that correlation is no proof of causation.
- The idea here is (1) to make the individual capable of controlling her entire data distribution and (2) thus reducing the amount of *potentially discriminatory* attributes leaking into arbitrary data silos.

1.2 Purpose & Outcome

- Personal Data Store (aka. Service, Space, Vault, Cloud, Management/Manager), VRM (Vendor Relation Management) aka "CRM upside down"
- Open Source

1.3 Terminologies

- serverless: https://auth0.com/blog/2016/06/09/what-is-serverless/
- digital footprints: TODO
- owner: person who controls (and probably hosts) the data service con-

taining her personal data

- (data) consumer: Third party, external entity requesting data, authorized by the owner to do so

2

Fundamentals

2.1 Personal Data in the context of the Big Data Movement

- difference between users *profile/account data* and their *meta data*?

2.2 Personal Data as a Product

...to do what exactly? (NOTE: was previously in *Motivation*)

- If one would ask collectors for their motivation to do so, most likely the answer would be something along the lines of *"We want to have a better understanding of our customers"*. To do what exactly?

- To predict what might be the next thing I am supposed to buy Or what things I probably would like to consume but most certainly not yet know of?

- Let's take a look at some examples. An advertising service uses tracking data for targeted advertising. The more data they collect about an individual, the more accurate decisions they are able to make on what ads they have to display are the ones the individual will click on and then will go all the way down to disclose a successful purchase. This makes the placed advertisement more valuable for google and therefore more expensive to the advertiser, because of a high precision. Or a streaming provider content recommendation is also based on heavy user profiling done by looking at her consumption history, tracked platform interactions and probably many more vectors. As always, it depends on the business model, but it seems to be consensual, that it all comes down to improving and enhancing the collector's product (NOTE: needs at least some empirical evidence), in order to satisfy the customers - and that on the other hand depends on who is seen as customers.

- Nevertheless if we change the perspective, a lot of great things can be achieved with the help of huge amounts of personal data, such as:

  - planing and managing human resources for situations, like e.g. big events where attendees might need some help [2]

  - predicting infrastructure workloads [http://ieeexplore.ieee.org/document/7336197/]
  - making more accurate diagnostics to improve their therapy [3]
  - finding patters in climate changes, which otherwise wont get revealed [4].

- individuals then get in role of selling/offering it's own data to those who were previously collecting them

2.3 Personal Data as of the Law

- who is the owner in what situation or under what circumstance?
- difference between "Personal Data" and the owner of that data?

2.4 Digital Identity

- identity defining data (e.g. history of personal ID card)

- with such a system a human being is represented by a non-physical abstraction of herself. Which essentially is a list of attributes, that are at least for legal and civil administration purposes important. Their values in total are unique and representing the corresponding human. Certain attributes hold unique values within it's own context, for example the *social security number*.

- Thus it's not necessary to know the values of all attributes in order to identify it's owner

- therefore its imported to not see it as a reduction of a living individual to some bits and bytes

- what will happen with her data service after a person died?

2.5 Use Cases (NOTE: maybe move to 100_introduction)

- package shipment after buying sth online
- social network accessing arbitrary profile data
- making an online purchase
- credibility (requesting credit permission) validation by a certain financial institution: accessing arbitrary data
- patient/health record
- care (movement) data

2.6 Related Work

TODO

2.6.1 Research

- openPDS/safeAnswer [http://openpds.media.mit.edu/]
- TAS3 aka ZXID aka Synergetics (lead arch Sampo Kellomäki also Co-Authored openPDS papers)
- Higgins [https://www.eclipse.org/higgins/]

- Hub-of-All-Things [http://hubofallthings.com/what-is-the-hat/]
- ownyourinfo [http://www.ownyourinfo.com]
- PAGORA [http://www.paoga.com]
- PRIME/PrimeLife [https://www.prime-project.eu, http://primelife.ercim.eu/]
- databox.me (reference implementation w/ the “solid” framework)
- Microsoft HealthVault
- Industrial Data Space (german research project mainly driven by Fraunhofer Institute)
- Polis (greek research project from 2008) [http://polis.ee.duth.gr/Polis/index.php]

2.6.2 Organisations

- Kantara Initiative (former “Liberty Alliance”) [https://kantarainitiative.org/]
- Open Identity Exchange [http://openidentityexchange.org/resources/white-papers/]
- Qiy Foundation [https://www.qiyfoundation.org/]

2.6.3 Commercial Products

- MyData [https://mydatafi.wordpress.com/]
- Meeco (killing the ad provider middle man) [https://meeco.me/how-it-works.html]
- RESPECT network [https://www.respectnetwork.com/]
- aWise AEGIS [http://www.ewise.com/aegis]

2.7 Standards and Specifications

- http(s)
- all the *Semantic Web* stuff
- Container/App spec
- JWT
- oAuth (?)
- JSON
- REST
- GraphQL

3

Core Principles

NOTE: here we discuss a variety of possibilities –> conceptual work

3.1 Data Ownership

- user-centric, full control

3.2 Identity Verification

- maybe go with a Signing/verifying Authority (aka CA)
  - do I trust the gov or certain companies more? Which interests do these Role/Stakeholder have?
  - revoking the cert which provides the authenticity of the individual's digital identity should only be possible with a two-factor secret. One part of this secret is owned by the CA and the other half has the individual behind the personal API
- TODO: look into
  - PKI
  - ePerso
  - E-Post/de-mail
- Authentication

3.3 Authorisation (remove in favour of data access?)

- NOTE: does not mean this tool authenticates it's owner against third party platforms like OpenID does. but it could play the role of the 2n factor in a multi-factor authentication process (if the mobile-device-architecture was chosen)
- refers primarily to the process of a data consumer (third party, which needs the data for whatever reason) verifies her admission to request

3.4 Authentic Data

- is this data (in this case identity) certified or not (results in higher value)

3.5 Supervised Data Access

- pure/plain data request/resonse
- remote computation/execution (assuming there is no client for the consumer) like https://webtask.io/

3.6 Encapsulation

- containerization (coreos, rkt, mirageos aka unikernal)

3.7 Open Development

- which and why open standards
- why open source
- collaborative transparent development
- Hosting & Administration
  - DYI
  - Usability

4

Requirements

4.1 User Interaction

- as effortless as possible

4.2 Management & Organisation

4.3 Administration

5

Design

5.1 Architecture

- showing possible directions, e.g.:
  - cloud or local storage
  - which components can go where
  - remote execution, to prevent data from leaving the system

5.1.1 Overview

- distributed architecture (e.g. notification/queue server + mobile device for persistence and administration)

5.1.2 Components

5.1.3 Plugins

- but for what? and not harm security at the same time? maybe just for data types and admin UI to display analytical (time based) data in other ways
- what about extensions (see iOS 10) to let other apps consume data; only on a mobile device and only if the data is stored there

5.2 Data

- keep in mind to make it all somehow extendible, e.g. by using and storing corresponding schemas

- NOTE: step numbers marked with a * are somehow tasks which are happening in the background and don't require any user interaction

5.2.1 Modelling

5.2.2 Categories (or Classes)

5.2.3 Types

5.2.4 Persistence

- database requirements

5.2.5 Access & Permission

- data needs to have an expiration date

IF01 - Authorizing a consumer to request certain data

1) owner creates a new endpoint URI (like *pdaas.ownersdomain.tld/e/consumer-name*) within the *management user interface*
2) owner passes this URI on to the *consumer*, e.g. through submitting a form or using any arbitrary, eventually insecure channel 3*) consumer need to call this URI for the fist time to verify it's authenticity
3) owner then get's a notification which asks her for permissions to access certain data under the listed conditions 5*) consumer will be informed about the outcome of the owner's decision (NOTE: alongside with some details? how do they look like? XXX need to be in the spec)

5.2.6 Consumption (data inflow)

- how data will get into the system
- hwo is the user able to do that, and how does it works

5.2.6.1 Manually

5.2.6.2 Automatically

5.2.7 Emission (data outflow)

- depending on what category of data, they might need to get anonymized somehow before they leave the system
- oAuth (1.0a and 2) requires consumers to register upfront. Since the current flow indicates that the initial step is done by the owner, that would cause an overhead in user interactions. Although the owner already *authorized* the consumer simply by submitting a unique URI (`pdaas-server.tld/register?crt=CONSUMER_REGISTER_TOKEN`), of which the `crt` is considered private. Even though the registration provides the consumer with mandatory information such as a consumer identifier (v1: `oauth_consumer_key`, v2: `client_id`) and, depending on the client type, a secret (see https://tools.ietf.org/html/rfc6749#section-2), this process it is not part the specification (https://oauth.net/core/1.0a/#rfc.section.4.2, https://tools.ietf.org/html/rfc6749#section-2). This enables the possibility of integrating oAuth into the consumer registration flow

by using the `CONSUMER_REGISTRATION_TOKEN` as oAuth's *client identifier*. The lack of credentials (v1: `auth_consumer_secret`, v2: `client_secret`) would require transferring the consumer identifier done over a secure channel (e.g. TLS). That would leave *oAuth2* as the version of choice, since it relies on *HTTPS* adn therefore makes the *secret* optional. Where on the other side oAuth 1.0a requires a *secret* to create a signature in order to support insecure connections..

- A general and URI for 3rd parties to register (aka requesting authentication) would raise the issue of dealing with spam request and how to distinct these from the actual ones.

5.2.8 History

- data versioning
- access logs

5.3 Interfaces

5.3.1 Internal

- UI for Management & Administration

5.3.2 External

- should there be a way to somehow request information about what data is available/queryable, or would this be result in spam/crawler and security issues (also a question for the topic of permissions/sensibility level of certain data)
- certain types of requests, depending on expire date:
  - "ask me any time"
  - "allowed until further notice"
  - one-time permission (but respecting certain http error codes and possible timeout - that might not count)

6

Specification

- what does *open* in Open Specification even mean?

6.1 Processes (TODO: find another word; “Protocol flows”?)

6.2 Application Programming Interfaces

6.3 Graphical User Interfaces

6.4 Security

- the downside of having not just parts of the personal data in different places (which is currently the common way to store), is in case of security breach, it would increase the possible damage by an exponential rate Thereby all data is exposed at once, instead of not just the parts which a single service has stored
- does it matter from what origin the data request was made? how to check that? is the requester’s server domain in the http header? eventually there is no way to check that, so me might need to go with request logging and trying to detect abnormal behaviour
- is the consumer able to call the access request URI repeatedly and any time? (meaning will this be stateless or stateful?)
- initial consumer registration would be done on a common and valid https:443 CA-certified connection. after transferring their cert to them as a response, all subsequent calls

need to go to their own endpoint, defined as subdomains like `consumer-name.owners-notification-server.tld`

6.4.1 Environment

6.4.2 Transport

- communication between internal components *must* be done in https only, but which ciphers? eventually even http/2?

6.4.3 Storage

6.4.4 Authentication

- how should consumer authenticate?

6.5 Recommendations

6.5.1 Software Dependencies

6.5.2 Host Environment

7

Conclusion

7.1 Ethical & Social Impact (TODO: or “Relevance”)

7.2 Business Models & Monetisation

- possible resulting direct or indirect business models
- owner might want to sell her data, only under her conditions. therefore some kind of infrastructure and process is required (such as payment transfer, data anonymization, market place to offer data)

7.3 Challenges

7.4 Solutions

7.5 Attack Scenarios

7.6 Future Work

- maybe enable the tool to play the role of an own OpenID provider?

7.7 Summary

- main focus
- unique features
- technology stack & standards
- resources
- the tool might be not a bulletproof vest, but

[1] D. Pedreshi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 560–568 [Online]. Available: http://dl.acm.org/citation.cfm?id=1401959. [Accessed: 03-Nov-2016]

[2] J. Ao, P. Zhang, and Y. Cao, "Estimating the Locations of Emergency Events from Twitter Streams," *Procedia Computer Science*, vol. 31, pp. 731–739, 2014 [Online]. Available: http://linkinghub.elsevier.com/retrieve/

pii/S1877050914004980. [Accessed: 05-Nov-2016]

[3] G. Palem, “The Practice of Predictive Analytics in Healthcare,” *ResearchGate*, Apr. 2013 [Online]. Available: https://www.researchgate.net/publication/236336250_The_Practice_of_Predictive_Analytics_in_Healthcare. [Accessed: 05-Nov-2016]

[4] N. Burger, B. Ghosh-Dastidar, A. Grant, G. Joseph, T. Ruder, O. Tchakeva, and Q. Wodon, “Data Collection for the Study on Climate Change and Migration in the MENA Region,” 2014 [Online]. Available: https://mpra.ub.uni-muenchen.de/56929/. [Accessed: 04-Nov-2016]