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

G. Jahn

January 13, 2017

Abstract

Data is the currency of tomorrow. Organizations, whether in the private or public sector, are gathering enormous amounts of personal (big) data. This data is harvested and incorporated by these third parties, but were created by individuals and should, therefore, belong to them. People are depending on their data. Their identity as well as their personality are defined by their personal data. Meanwhile data silo operators are hammering onto these haystacks eagerly trying to find any correlations worth interpreting, thereby almost inevitably discriminating against the rightful owners. To reduce the possibility of discrimination only bare minimum of data required should be handed over to a 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 would then also encourage self-hosting.

Contents

This page intentionally left blank

1

Introduction

1.1 Motivation

Nowadays it is rare to find someone that does not collect data about some kind of thing; particularly humans are the targets of choice for the *Big Data Movement* [1]. Since humans are all individuals, they are - more or less - distinct from each other. However, subsets of individuals might share a minor set of attributes, but the bulk is still very unique to an individual, given that the overall variety of attributes is fairly complex. That small amount of shared attributes might seem to be less important, due to the

nature of inflationary occurrence, but the opposite turns out to be true. These similarities allow to determine the individuals who are part of a subset and the ones who arn't. Stereotypical patterns are applied to these subsets and thus to all relating individuals. Thus enriched information are then used to help predicting outcomes of problems or questions regarding these individuals. In other words, searching for causation where in best the case one might find correlations - or so called *discrimination*, which

> [...] refers to unfair or unequal treatment of people based on membership to a category or a minority, without regard to individual merit. *[2]*

When interacting directly with each other, discrimination of human beings is still a serious issue in our society, but also when humans leverage computers and algorithms to uncover formerly unnoticed information in order to include them in their decision making. For example when qualifying for a loan, hiring employees, investigating crimes or renting flats. Approval or denial, the decision is based on computed data about the individuals in question [3], which is simply discrimination on a much larger scale and with less effort - almost parenthetically. The described phenomenon is originally referred to as *Bias in computer systems* [4]. What at first seems like machines going rouge on humans, is, in fact, the *cognitive bias* [5] of human nature, modeled in machine executable language and made to reveal the patterns their creators were looking for - the *"Inheritance of humanness"* [6] so to say.

In addition to the identity-defining data mentioned above, humans have the habit to create more and more data on a daily basis - pro-actively (e.g by writing a tweet) and passively (e.g by allowing the twitter app accessing

their current location while submitting the tweet). As a result, already tremendous amounts of data keep growing bigger and bigger, waiting to be harvested, collected, aggregated, analyzed and finally interpreted. The crux here is, the more data being made available [7] to *mine*, the higher the chances to isolate data sets, that differ from each other but are coherent in themselves. Then it is just a matter of how to distinguish the data set and thereby the related individuals from each other.

In order to lower potential discrimination we either need to erase responsible parts from the machines, thereby it's crucial raising awareness and teaching people about the issue of discrimination, or we try to prevent our data from falling into these data silos. The latter will be addressed in this work.

1.2 Purpose & Outcome

From an individual's perspective providing data to third parties might not seem harmful at all. Instead eventually one get improved services in return, e.g. more adequate recommendations and fitting advertisement, or more helpful therapies and more secure environments. That said, though it is a matter of perception what's good and bad, what's harmful and what's an advantage. Computing data to leverage decision making is essentially just science and technology and it's up to the humans how such tools are getting utilized and what purposes they are serving. Hence it should be decided by the data creators, how their data get processed and what parts of them are used.

To tackle the described issue the initial idea here is (1) to equip individuals with the ability to control and maintain their entire data distribution and (2)

thus reducing the amount of *potentially discriminatory* [2] attributes leaking into arbitrary calculations. To do so people need a reliable and trustworthy tool, which assists them in managing all their *personal data* and making them accessible for 3rd parties but under their own conditions. After getting permission granted these data consumers might have the most accurate and reliable one-stop resource to an individuals's data at hand, while urged to respect their privacy at the same time. However this also comes with downsides in terms of security and potential data loss. Elaborating on that and discussing different solutions will be part of the [design process][Design].

The way how to solve the described dilemma is not new. Early days of work done in this field can be dated back to the Mid-2000s where studies were made e.g. about recent developments in the industry or user's concerning about privacy, and the term *Vendor Relationship Management (VRM)* were used initially within the context of user-centric personal data management, which also led into the *ProjectVRM* [8] started by the *Berkman Klein Center for Internet & Society at Hardvard University*. Since then a great amount of effort went into this research area until today, while also commercial products and business models trying to solve certain problems. For instance concepts such as the *Personal Data Store (PDS)* [9] or a *MyData* [10] implementation called *Meeco* [11], which will all be covered in a more detailed way within the following chapter.

The work and research done for this thesis will be the foundation for an *Open Specification*, which by itself is a manual to implement a concept called *Personal Data as a Service*. Important topics like how the architecture will look like, where the actual data can be stored, how to obtain data from the ex-

ternal API or what requirements a user interface for data management need to satisfy, will be examined. After the thesis will be finished, the majority of core issues should already be addressed and can then get outlined in the specification document. Only then the task to actual implement certain components can begin. The reason for that is, when sensitive subjects especially like people's privacy is at risk, all aspects in question deserve a careful considerations and then get addressed properly. Thus it is indispensable to put adequate effort primarily into the theoretical work. To be clear though, that doesn't mean writing code to test out theories and ideas can't be done during research and specification development. It might even help to spot some flaws and eventually trigger evolvement.

To ensure a great level of trust to this project and the resulting software, it is vital to make the development process fully transparent and encourage people to get involved. Therefore it is required to open source all related software and documents [12] from day one on.

In summary, this document is meant to be the initial step in a development process fabricating a tool to manage all data defining a data subject's identity, that is controlled and administrated by that individual, so that maybe she is giving a more precise understanding about where her personal information flows and how this might effect her privacy.

1.3 Scenarios

The following use cases shall depict different situations and possible ways such emerging software might be applicable or useful, while providing it's

user with more control over her personal data. Some of them are more practical and realistic, like ordering and purchasing online a product, others might have no current usage, but showing a certain potential to become more relevant when new technologies and business models emerge, followed by new demands of data.

- order sth online, purchase, and package shipment
- social network accessing arbitrary profile data
- credibility (applying for a loan) validation by a certain financial institution: accessing arbitrary data
- patient/health record
- care (movement) data

1.3.0.1 Ordering a product online

The data subject searches through the web to find a new toaster, since her old one recently broke. After some clicks and reviews, she found her soon-to-become latest member of the household's kitchenware. After putting the model name in a price search engine, hoping to save some money, the first entry, offering a 23% discount, caught her attention. She decides to have a deeper look into the toasters and thus has heading towards the original web shop entry. Finally she came around and put the item onto her card, despite tha fact, that she has never bought something from that online shop before. Then she proceeded to checkout to place her order. The shop-interface is asking her to either insert her credentials, proceed without registration or sign-in, or insert s URI to an endpoint of her *Personal Data as a Service*. TODO: the following description might need some adjustments according

data flow / process description She opens up the management panel of her *PDaaS* and creates a new entry in a list of data consumers, that already have access to characteristics of her personal data. As a result, she receives a URI, which she inserts according, as mentioned before; after she assures herself that the data exchange with the shop through the browser is based on a secure connection (HTTPS). Under this URI, the shop-system can then request data, that is required for a successful transaction. Moving on to the next step after submitting the URI, the data subject is ask to decide how she would like to pay. The choices are: credit card, invoice, online payment provider of choice or bank transfer. She chooses the last one, submits her selection and thereby completes her order. After a moment, a push notification pops up on her mobile device, which is a permission request from her *PDaaS*, asking for granting the shop-system, she just places the order, access to her full name, address and email. Additionally she can decide between three states of how log the permission wil be valid: *one-time-only*, *expires-on-date* and *until-further-notice*. Since she never ordered at this shop before and might never again, she decided to grant access only for this specific occasion. After the shop-system receives the data, it sends an email to the data subject, containing some information about her order, including the shop's bank details. which then enables her to actually pay the amount due. After the system recognizes the payment has coming in, it triggers the shipment of the toaster. In order to get a full impression of how the whole process might have look like when the data subject had chosen one of the other payment methods, the differences will be describes in the following. If the data subject would have wanted to pay with her credit card, the only difference would have been, that the shop-system had requested

also to access the credit card number and it's belonging secret, and when sending the email the system would have omitted the information about the shop's bank details. Being able to choose paying with invoice where possible only because the *PDaaS* response has indicated, that it's containing *profile data* is certified and therefore trustworthy. Which reduces the shop owner's risk and would have enabled him in case of fraud or misuse to take action. Choosing to involve paypal as a *middleman* to process the payment, requires the data subject to had already granted paypal certain access to her *PDaaS*. If that's the case, then the shop-system would have ask also for her paypal-ID, which then the system will use to request the payment directly from paypal. This on the other side will cause paypal to consult the *PDaaS*, which results in a second notification, asking the data subject for permission to proceed. After the payment transfer was successful, the shipment will gets initiated. And with the package arriving at the data subject's doorstep the whole transaction has finished.

1.3.0.2 Interacting with a social network

Entering a social network for the first time, only take the URI to the data subject's *PDaaS* and a password. The data subject receives a notification on her mobile device asking for permission to access certain data about her. If her mobile device is currently not at hand, she can also use the administration panel provided by her *PDaaS* and reachable with a web browser on every internet-enabled device. Within that panel pending permission reviews will be indicated. Whether the data subject has already reviewed the request or not, she should be able to login to the social network. After doing

so, she should not be able to see any of her information. After granting permissions to the social network to accessing certain data *until-further-notice* and reloading the session, she then should see all her So every time, someone on that network tries to access her information, whom she has allowed to see that information (which is managed by the user only from within the network), the network pulls the data from the data subjects's *PDaaS*, if it's still permitted to do so. It's also imaginable, that the social network and a *PDaaS* are establishing a backward channel. This channel could be used to send all the content she would create over time while interacting with the social network and it's participants back to her *PDaaS*. The network itself only stores a reference to all content object, whether it's for example an image, a post or comment on somebody else's post and if it's needed the actual content will be fetched from the data subject's *PDaaS*.

1.3.0.3 Applying for a loan and checking creditworthiness

The data subject would like to buy an apartment. In order to finance such a acquisition, she needs a funding, which in her case, will be based on a loan. During a conversation in a credit institute of her choice, an account consultant describes to her what data will be required in order to decide about her creditworthiness. While giving a consensual nod, she takes out her smartphone and brings up the management panel of her *PDaaS*. With a few taps she has just created a new *data consumer*. The panel then shows a QR-Code, that holds a URI to a dedicated endpoint of the data subject's *PDaaS*. She shows that code to her consultant, who then scans it. While

handling some more formalities and talking about several issues and possible products she might be interested it, she gets a notification on her phone, informing her about a permission request the institute just made. It lists all the different data points the institute would like to access in order to calculate her scoring, such as address, monthly income, relationship status and family, history of banking or other current loans. After some back and forth and solving some misunderstandings with the help of her consultant, she decided to just partially allow access to the requested data and just for this time and purpose. The consultant kindly pointed out, that these decisions might have an impact on the scoring and thereby on the lending and it's terms. After the consultant got a signal from the computer system, the two then finishing up their meeting and the consultants informed the data data subject about the next steps, which includes a note, that the institute will contact her within the next few days, when they have come to a conclusion. In case of a positive outcome a new appointment need to be made, for doing all the paperwork and signing the contract. From a technical point of view, two different ways of computing the score are imaginable. The first one would be, transferring only the plain data - request, containing the query and response containing the data - including the expire date and information regarding the signature state. But the actual computations and analytics to obtain the score, will happen within the infrastructure of the credit institute. When this process is over, all transferred personal data has to be deleted. An alternative could prevent the data from leaving the *PDaaS*, in which the institute's request won't consists of a data query. Instead it would came along with an chunk of software and some information on how to run it. The *PDaaS* server will provide an isolated runtime in which the software then gets executed. After

the process has finished, the result will be send back to the credit institute's infrastructure.

1.3.0.4 Maintain and provide it's own health/patient record

Some time ago on a hiking trip in a moment of carelessness the data subject has accidently broke her leg. She came into a hospital and went straight into surgery, where the physicians could fix the injury. Time went by and the leg has healed completely. After she woke up today she felt some pain coming from that area where her leg was broken. She decided to call in sick and went straight to a doctor nearby. During her recovery she visited that doctor regularly. At the reception desk, she opens up the *PDaaS*'s management panel on her smartphone and searched through the list of data consumers. After she found the entry for this clinic, she flipped her phone to show the receptionist the corresponding QR-Code, which she started to scans immediately. However the receptionist couldn'd see any data on the screen, because the access has already expired. The data subject only had permitted access for the estimated time of recovery, which was over some time ago. That's why she got a notification, to re-grant some access. Going through the data points the clinic-system has requested, she noticed that her address is incorrect. Last month she moved out and into a bigger apartment just down the street. She must have forgot to change that data, which she corrects immediately right before submitting the access configurations for the clinic-system. She also included the access to all the data originated from that time after her accident. A moment later the receptionist confirms

to now being able to see all necessary data. The data subject takes a seat in the waiting room. While passing some time, she had a deeper look into her list of data consumers; some of them she couldn't even remember and for others she was surprised to what data she has granted access to and started to reduce certain permissions, if it was appropriate in her eyes. She even removed some of the entries. The appointment with her doctor went great. He even had to review the x-ray images in order to make a adequate differential diagnosis. After the visit, she had to make a quick stop at a pharmacy along the way to pickup the drugs her doctor had prescribed for her to reduce the pain. She had to wait in the queue with two other customers being in front of her. She realized, that it's the first time she has been here. So she prepared a new entry in her data consumer list, including all information about her prescriptions. So by the time she get served, she just let the person behind the register scan her code. In the next seconds the data subject gets a quick confirmation notification about the request that just happened. A moment later the pharmacist come back with her drugs, which she then pays in cash and the transaction is done.

1.3.0.5 Vehicle data and mobility

Assuming a car itself has no hardware on board in order to establish a wireless wide area connection to an outside access node. Only from the inside one can connect to the car (wired or wireless). After entering a car, on the data subject's mobile device pops up a notification asking for permission to connect to that device. In addition to the expiration date, the data subject can choose to en- or disable two more options. First, a wifi network with an

uplink to the internet can be provided to everyone inside the car. Secondly, connections, the car might want to establish, in order to emit data via internet - which, regardless, have to go through the currently linked mobile device. Thus the device owner gains full control over any external data transfer that might happen. This again would allow two things: (A) permission management for all outgoing data and (B) funnel all data generated and provided by the car into the *PDaaS* associated with that linked device. It might also be feasible to deny any connection the car is trying to make. Thus the data will only be stored in the *PDaaS*. If somebody is interested in such then have to ask for access permission. That same concept about movement tracking and vehicle data could also be applied to driving (motor) bicycle.

1.4 Terminologies

Web Service TODO

Open Specification TODO

Big Data deep learning, neural networks

Profile Data individual's inherent data; TODO

Personal Data (TODO) Personal Information predominantly static data points related to an individual

Personal Data as a Service (PDaaS) a web service controlled, owned and maybe even hosted by an individual, that provides access to the data subject's personal data and offers maintainability as well as permission management. It can be seen as her personal agent; sometimes also referred to as *the system*

Personal Data Store TODO

Vendor Relationship Manager The *ProjectVRM* defines the a VRM as follows: "TODO" [13]

Personal Information Management Systems (PIMS) TODO [14]

serverless TODO https://auth0.com/blog/2016/06/09/what-is-serverless/

Digital Footprints TODO

Data Subject an individual who first and foremost is the owner of all of her personal data; sometimes referred to as *owner*

Operator a *data subject* using a PDaaS to control (and probably host) her personal data; sometimes referred to as *data controller*

(Data) Consumer Third party, external entity requesting data, authorized by the data subject to do so; sometimes referred to as *(data) collector*

Data Broker(s) entities with commercial interests, that collect, aggregate and analyze information/data of any kind - in this case about human beings - from different sources in order to enrich the data sets, to finally license the resulting corpora to other organisations. [15]

Permission Request fist attempt to request access to certain data in the *PDaaS*

Access Profile a data set about a third party that already made an permission request. The set contains additional information and access rules

Data Access after a third party's *permission request* got reviewed and saved, that entity is then able to make an attempt to access data.

2

Fundamentals

The following chapter shall provide the foundational knowledge about concepts like *Personal Identity* or *Big Data* and therefore ensures a common understanding on their relation to the problem this work tries to solve. Additionally it is given a brief overview on what existing standards and technologies might be used, and summarizes the research already been made as well as it's current state.

2.1 Digital Identity, Personal Data and Ownership

- *Digital Identity*
  - what is a *DI*? and in comparison to *Personal Data*?
  - what is required to make the PDaaS used or seen as a *DI*?
- *Personal Data* definition
  - general - freely spoken
  - as of EU law (incl citation)
  - as of US law (incl citation)
  - is it just policy/guideline or enforceable too (law/rule)? what relevance/impact have companies *terms and conditions*?
  - EU and USA (since server might be located outside the state or effective range)
- *Ownership* of personal data
  - who is the owner in what situation or under what circumstances?
  - am I the owner when I was the one who was collecting them? Does it depend on whether the resource was public or somewhat private?
  - what will happen with her data service after a person died?

- A **Digital Identity** is a non-physical abstraction of an entity, such as an organisation, an individual, a device or even software, which allows bidirectional association. In the context of this document, it only refers to human beings. Therefore a *digital identity* is the individual's representation in digital systems, consisting of identity-defining data, such as *personal information* and it's history and preferences [16]. *Personal information*, in this case, refers to inherent (date of birth) and

imposed (credit card number) characteristics.

- From a technical perspective a DI is essentially a collection of characteristics, attributes and time series data (e.g. interaction logs or bank account history). A subset of these attributes combined can form unique fingerprint, like certain single data points (e.g. social security number) in their own context might be, too. Thus it might not be necessary to know the values of all attributes in order to identify a person as the rightful owner and physical counterpart. It can also be seen as an avatar in the digital world or as the digital part of a human's identity. Therefore its important to not view the *DI* as a reduction of a living individual to some bits and bytes, but rather as a appropriate representation for certain purposes and contexts.

- It is also possible to provide an additional level of authenticity insurance for data related to an entity. Therefor an unrelated third party, which needs to be approved not only by the related individual, but also by all entities participating in a context, which might be relevant e.g. for some administration purposes.

- But the concept would also impose a new level of attacking vectors to the identity owner, such as identity theft. The attacker is no longer required to be physically present to be able to steal certain unique identifiers from a person. It is sufficient to gain access to area where the sensitive data is stored.

- In the context of this document and all related work, **Personal Data** is specified as a combination of an individual's *Digital Identity* and all of it's ever created intellectual property [17] (e.g. posts, images, tweets

or comments). This includes all sorts of tracking data and interaction monitoring, as well as metadata manually or automated enriching content (e.g.geo-location attached to a tweet as meta information). Data, captured by someone ore something on or about the individual's private living space and property. Simply every data point reflecting the individual's personality - partly or as a whole - is seen as *personal data*.

- The european *Data Protection Regulations* defining *Personal Data* as follows: > 'personal data' means any information relating to an identified or identifiable natural person > ('data subject'); an identifiable natural person is one who can be identified, directly or > indirectly, in particular by reference to an identifier such as a name, an identification > number, location data, an online identifier or to one or more factors specific to the physical, > physiological, genetic, mental, economic, cultural or social identity of that natural person; > *[18]*

- The U.S.A. has little legislation on defining and protecting consumer's privacy. At least they have no explicit bills addressing such area [19]. Though some of the existing sectoral laws consist of partially applicable policies and guidelines [20]; most of them addressing specific types of data. In 2015 the White House made an attempt to fill the gap with the *Consumer Privacy Bill of Rights Act*, but to this date it didn't passes the draft state. According to the critics, it lags of concrete enforceable rules consumers can rely on [21]. The draft contains a general definition of *Personal Data*: > "Personal data" means any data that are under the control of a covered entity, not otherwise > generally available to the public through lawful means, and are linked,

or as a practical matter > linkable by the covered entity, to a specific individual, or linked to a device that is > associated with or routinely used by an individual, including but not limited to [...] > *[22]*

- followed by a list of concrete data points, e.g. email or postal address, name, social security number and alike. Aside from the legislation with bills, a few third-party organisation can also participate by and add new or overwriting existing rules and policies. Namely for example the *Federal Communications Commission* (FCC), recently releasing *Rules to Protect Broadband Consumer Privacy* including a list of categories of sensitive information [23], which wants *Personally Identifiable Information* (alias Personal Data) to be understood as: > [...] any information that is linked or linkable to an individual. [...] information is > "linked" or "linkable" to an individual if it can be used on its own, in context, or in > combination to identify an individual or to logically associate with other information about a > specific individual. > *[24]*

- Despite minor difference in detail, they all have similar ideas of personal data and their belonging. Even though, the version proposed by EU is almost identical with the definition introduced for the context of this work. Although the FCC's statutory authorities might be somewhat debatable regarding certain topics, the *Communications Act* as a U.S. federal law equips the FCC with power to regulate and legislate.

- Having a common opinion on what data points are belonging to person is the foundation to define a set of rules on how deal with *Personal Data* accordingly. Every business, operating within the EU, is required[1]

[1]according to article 12-14 of the *EU General Data Protection Regulation 2016/679*

to provide it's users with a *Privacy Policy*, while e.g. in the U.S. - as mentioned above - only partially and depending on context and data type users must be informed about which and how their data get processed [25].

- A user commonly agrees on the privacy policy, by starting to interact with the author's business, thus every *Privacy Policy* is required to be publicly accessible; e.g. before creating an account. > By clicking Create an account, you agree to our Terms[2] > and that you have read our Data Policy[3], including > our Cookie Use[4]. > *[web_2016_facebooks-landing-page_policy-acknowledgement]*

- It can be seen more likely an information notice, that translates and specifies general given law, rather then a contract.

- With such knowledge at hand, it is up to each individual, if the service's benefits are worth sharing some personal data, while simultaneously acquiescing potential downsides concerning the privacy of such data.

- Every entity who is doing so, muss process Personal data according to the law and their *Privacy Policy*. If they policies are violating existing law or the entity effectively goes against the law with their actual doing, penalties might follow. Depending on the level and impact of their infringement in addition the law itself, aside from revising their wrongdoings the entity might have to compensate the affected individuals, pay a fine or get revoked their license.

[2]https://www.facebook.com/legal/terms
[3]https://www.facebook.com/about/privacy
[4]https://www.facebook.com/policies/cookies/

- Not only privacy laws, but every legal jurisdiction has it's limitations - concerning their territorial nature - which makes legislation not exactly an appropriate tool when it comes to fixing existing issues and strengthen the individual's privacy and rights in a global context like the *world wide web*. If no international agreement is in place [26], only those laws are considered valid and enforcible where the organisation is registered, and maybe the fact where (meaning in which area of jurisdiction) the their servers are located or the data is processed and stored.

Whereas **Ownership** of *Personal Data* has no legal ground foundation what so ever. The concepts of intellectual property protection and copyright might intuitively be applicable, because the data, created by the data subject, seems to be her *intellectual property*. Such property implies to be a result of a creative process though, but unfortunately there is no *threshold of originality* in facts, like *personal information* is [27].

- Ownership in the sense of having exclusive control over it's personal data and how they get processed at any given point in time; this not only comes with high costs, but is also very inconvenient for both parties - data subject and data consumer. It consists of two concepts: (A) the right to do what every is desired with their property and (B) in which rules and mechanisms the ownership can be assigned to someone [28].

- The european DPR[5] contains only one occurrence of the word *ownership*, which is not even related to the context of *personal data* or the

[5]EU Data Protection Regulation

data subject. It only stats, that *"Natural persons should have control of their own personal data."* [29]. Whereas Commissioner J. Rosenworcel of the FCC wants *"consumers [...] to [...] take some ownership of what is done with their personal information."* [30]

- Typically the question of data ownership is addressed in data consumer's *Terms of Service* (ToS), which an individual might have to accept in order to establish a (legal) relationship with it's author. I should be kept in mind, that *ToS* might change over time; not necessarily to the users advantage. All addressed issues (by the ToS) must not violate any applicable or related law, otherwise the *ToS* might not be legally recognized. Taking the following excerpts from different *ToS*:

  > You own all of the content and information you post on Facebook, and you can control how it is shared [...]. *(under "2. Sharing Your Content and Information", by Facebook [31])*

  > You retain your rights to any Content you submit, post or display on or through the Services. What's yours is yours — you own your Content. *(under "3. Content on the Services", by Twitter [32])*

  > Some of our Services allow you to upload, submit, store, send or receive content. You retain ownership of any intellectual property rights that you hold in that content. In short, what belongs to you stays yours. *(under "Your Content in our Services", by Google [33])*

  > Except for material we may license to you, Apple does not claim

ownership of the materials and/or Content you submit or make available on the Service "(under"H. Content Submitted or Made Available by You on the Service", by Apple [34])*

All these statements are followed by the same term, stating that the user grants the author a worldwide license to do almost any imaginable thing with her data. This even applies to Apple, if the user is *"submitting or posting […] Content on areas of the Service that are accessible by the public or other users with whom [the user] consent to share […] Content"* [34].

- It is worth noticing, that in every *ToS* it is only referred to the data subject's content, not all her personal data. As mentioned above, personal information are no intellectual property, but playing an important role in data analytics though. Which is why *privacy policies* are in place, to ensure at least some user enlightenment, even though it doesn't compensate the lack of control.

- In addition to that, the meaning of *ownership* used in the quoted *ToS* is missing a clear outline and thus causing ambiguity and leaving room for interpretation. Nor the actual definition of *ownership*, as described earlier, is applicable for these kind of cases, since the user losing all its control is by design. Handing over data to the consumer annihilates the exclusive control over the data and revokes the ability of assigning such control. There is no (legislation based) way to establish a feasible concept of *ownership*, if the data consumer has no motivation to promote the user the a comprehensive owner of her data.

- Leaving all the legal layer aside for a moment and switching the perspectives a bit; Data consumers might argue, that they had invested

in enabling themselves to collect, process and store personal data, so it belongs to them. But from the data subject's point of view it might only be the case as long as as she would benefit as well somehow, e.g. using products, services or features, offered by consumers, which quality depends on personal data. If the data subject chooses to move to a competitor might what to bring her personal data with her. But then again the former data consumer would object, competitors would benefit from all investments the consumer has made, but without any effort. Though, not entirely wrong, two aspects need to be emphasize. (A) In order to archive a high level of quality for their analytics and therefore in making right decisions to gain improvement, it's vital to huge amount of effort in developing these underlying technologies, not only in acquiring personal data. Which again only constitutes (B) the foundation of various subsequential computations followed by an ongoing collecting, aggregation and analytics of actively and passively created data and metadata (e.g. food deliver history or platform interactions and tracking). Given the initially introduced definition of *personal data* it appears to only be a fraction of the involved data belonging to its owner. The larger part consists of highly valuable metadata [35] [36] and therefore should remain to the data collector and either be deleted or sufficiently anonymized, if the owner cancels the relationship. The data subject should not depend on the collector's willingness when it comes to handing over her personal data (e.g. list of favorites or delivery history). Instead, using her own tool to provide the consumer with required data (e.g. list of favorites) or tap into her data creating interactions (e.g. food deliveries) on her own.

- Whether an individual dies or a user deletes her account, as long as certain data point are shared with / connected to other users, the data will remain. At least when it comes to facebook.
- Generally speaking, all data solely associating with an individual, is in the ownership of the same. But since it doesn't exist any legal concepts on *personal data* ownership, a technical solution could help to regain some control.

2.2 Personal Data in the context of the Big Data Movement

- big data itself initially can be seen as a *huge blob of data* containing more or less structured data sets [37], whose size might have exceeded the capabilities of retrieving certain information almost only by hand. Such high data haystacks usually come along with new challenges in logistic and resource management, when information retrieval needs to get automated on a large scale [38]. Theses practices are commonly referred to *Big Data (Analysis)* including distributed computing and machine learning.
- Big Data, or to be more precise, collecting and analyzing big data, serves the prior purpose to extract useful information, which on the other hand depends on what was the opening question about, but also what data sets the corpus is containing.
- At first, (A) formalizing question(s) that the results have to answer.

Secondly, (B) deciding what data is needed and appropriate and then start collecting. Third, (C) designing data models accordingly and correlate with the data (D) next, analyse and interpret the results. (E) last but not least, make business decisions based und the analyses ([39] Fig. 3).

- machine learning/data mining –> computers trained to find coronations

- since quite a few businesses (in terms of purpose or intention) are based around the concept of customers, which are generally somewhat entities consisting of at least one human being, personal data takes a major part in what *Big Data* can be about. In the context of this thesis, these entities are individuals with a unique identity. And to understand the behaviour, decision making and needs of her customers a vendor, who owns the business, needs to know as much as possible about them, when she wants to know what changes she needs to address in order to move towards the most lucrative business.

- personal data and information are reflecting all this knowledge. It starts with profile (or sensitive) data, such as gender, age, residency or income, goes on with time series events like geo-location changes, or web search history and goes all the way up to health data and self-created content like *Tweets*[6] or videos.

- all these classes of personal data hold a major share[7] in the field

[6]public massages published by an account on twitter.com, which will be displayed in the timeline of all her subscribers and also might contain additional types of content like images, links or video

[7]it doesn't matter whether an individual or just someone on behalf of an organisation spend money for something. at the end of the day, they are all humans on this planet and

of data analytics (TODO: find statistics showing shares of data types/classes/categories, [40] [41])

- but, depending on the specific attributes, they might be not that easy to acquire. in general most businesses obtain data from within their own platforms. some data might be in the user's rang of control (e.g. customer or profile data), but most of the data comes from interacting directly (content creation, inputs) or indirectly (transactions, meta information). the level of sensitivity is mainly based on the purpose of the platform (benefit for the user) and what is the provider's demand from the users commitment (e.g. required inputs or usage requires access to location)

- from a technical perspective collecting passively created data is as simple as integrating logging mechanisms in the program logic. since the industry moved towards the cloud[8] most scenarios utilized server-client architectures. Furthermore the *always-on* philosophy evolved to an imperative state. standalone software is starting to call the author's servers from time to time, just to make sure the user behaves properly. For browsers it was already a common narrative to make here and then requests to the server - still preventable though, but when it comes to native mobile apps it is almost impossible [42] to notice such behaviour and therefore preventing apps from doing so.

in a capitalistic oriented world money needs to flow and profits needs to be maximized. So to know where it will flow or why it will flow in a certain direction it is crucial to know everything about it's decision maker - the humans on this planet.

[8] side note - one might come to the conclusion, that only the trend towards the *cloud* made it actually possible to collect to such an extent we are all observing these days, because standalone software should not necessarily require internet connection and therefore the vendors had no way to gather information whatsoever

- these architectural developments were inducing the gathering of potentially useful information from all over the system on a large scale [43]. Logging events, caused by the user's interactions, on the client, which then get forwarded to backend servers. Or keeping track of all kinds of transactions, which is done directly in the backend. Before running together in a designated place, all these collected chucks of data (TODO or "data points") are getting enriched with meta information. Finally get stored and probably never removed again - all for later analyses.

- The mindset in the *Big Data Community* is grounded on the basic assumption of *more data is more helpful*, which already is emphasised by the often-cited concept of the three *Vs* (Volume, Velocity, Variety) [44]. which is not entirely wrong, because it lies in the nature of pattern and correlation discovery, to provide increasing quality results [45], while enriching the overall data with more precise data sets. But when new technologies are emerging, questioning the downsides and possible negative mid- or long-term impacts are typically not very likely to be a high priority. The focus lies on e.g. trying to to reach and eventually breach boundaries while beginning to evolve. So non-technical aspects such as privacy and security awareness doesn't come in naturally, instead a wider range of research needs to be done alongside the evolution process and the increasing adoption rate in order to uncover such issues. Only then they can addressed properly on different levels - technical, political as well as social. So that the *Big Data Community* itself is able to evolve, too. All in all it's a balancing act between respecting the user's privacy and having enough data at hand to satisfy the initial questioning with the computed results. Therefore people working in

such contexts need to have advanced domain knowledge, be aware of any downsides or pitfalls and need to be sensible about the ramifications of their approaches and doings. Such improvements are already happening, not only originating from the field's forward thinkers [46], but also advocated by governments, consumer rights organisations and even leading Tech-Companies start trying to do better [47] [48] [49] - as discussed in the section [TODO see personal data as of the law],

- earlier in the text a difference was made between actively created and passively created data
- based on that one could say *profile/account data* is actively created, because it got into the system by the user's actively made decision to insert these information into a form and submit it - for whatever reason. whereas detecting the user's current location and adding this information to the submitted form is *meta data*
- of cause, it is debatable whether these kind of data belongs, in the sense of being the rightful owner, to the user or to the author or owner of the software containing the code that effectively created the data.
- maybe personal data is every data/information whose creation (or digital existence) is a direct result of user interaction/engagement?
- lets have a look into what the rule book says about that –> next topic (law)

2.3 Personal Data as a Product

- *Big Data Analytics* by itself just comprises a structured and technical-aided procedure, serving the purpose of finding invisible information, that might be helpful to make (right) (business) decisions. Though, if one would ask data collectors about their motivation, most likely the answer would be something along the lines of PR phrasing like *"We want to have a better understanding of our customers"*. But 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 information they have about an individual, the more accurate decisions they are able to make about what ads are the ones the individual most likely will click on and disclose with a successful purchase. As a result this makes the placed advertisement more valuable for ad service and therefore more expensive to the advertisers, because of a high precision. Or a streaming provider's content recommendation is also based on heavy user profiling done by looking at her consumption history, tracked platform interactions and probably many more vectors. Another example is *Google Traffic* [50] [51], a service, integrated as a feature in *Google Maps*, which is Google's web mapping service. *Google Traffic* visualises real-time traffic conditions, when using *Maps* as a navigation assistant, to provide the user with a selection of possible paths, but enriched with duration, that takes such conditions into account. The

data, required to offer these information, is supplied by mobile devices, constantly sending GPS coordinates with a timestamp into Google's infrastructure. This, however, only is made possible, because Google's services are widely used in addition to the fact that the majority of mobile devices [52] is driven by Android, an mobile operating system developed by Google, that deeply integrates with it's services. For this case the same assertion can be made - the more constantly streaming geo-location data, the more precise the information are about traffic conditions. Since this information demands the real-time aspect, adding time to the equation, add a other dimension of complexity to problem.

- while the impact on our society of this first example group might be doubtable, a change of perspective opens up a different range of application areas. Such as
  - planing and managing human resources for situations, like e.g. big events or emergency situations where attendees might need some help [53]
  - predicting infrastructure workloads [TODO http://ieeexplore.ieee.org/document/7336
  - making more accurate diagnostics to improve their therapy [54]
  - finding patters in climate changes, which otherwise wouldn't be detected [55].
- Through all these examples, some of them might not necessarily founded on personal data, whereas others primarily depend on them and yet others only implicitly rely on data collected from individuals. As always, it depends on the purpose - also known as *business model*

- but it seems to be consensual, that it all comes down to improving and enhancing the collector's product in order to satisfy the customers - and that on the other hand depends on what is meant to be the product and who is seen as customers.

- Putting a top 10 list of industries using utilizing *Big Data* [56] right next to visualization showing categories of personal data targeted by data collectors

  [57], at least 7[9] of these industries can be identified as data collectors, whereas less then a half[10] are taking part of being a *Data Broker*, but almost all of them are using people's personal data, whether collected by themselves or acquired from *Data Broker*.

- At this point it's save to say, that *Personal Data* is either seen directly as a product, especially from a Dater Broker's point of view, or indirectly due to it's essential part in *Big Data* practices. The former generates direct revenue by selling these data and the latter might affect a business's product quality in a positive manner and thereby increasing revenue as well.

- At the end it all comes down to understanding the human being and why she behaves as she does. The challenge is not only to compute 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 by far no proof of causation.

[9] Banking and Securities; Communication, Media & Entertainment; Healthcare Providers; Government; Insurance; Retail & Wholesale Trade; Energy & Utilities

[10] Banking and Securities; Communication, Media & Entertainment; Insurance; Energy & Utilities

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

2.4 Related Work

The idea of a digital vault, controlled and maintained by the data subject, the individual, isn't that new. Holding her most sensitive and valuable collections of bits and bytes, protected from all these data brokers and authorities, while interacting with the digital and physical world, opening and closing it's door from time to time, to either put something important for her inside or retrieving an information important for someone else. While in the mid and late 2000s the growth of computer performance and capacity were crossing it's zenith (see Moore's Law [58]), at the same time the internet was starting to become a key part in many people's lives and in society as a whole. Facilitated by these circumstances, *cloud computing* has been on the rise, causing the shift towards parallel distributed processing and patterns alike. Thereby making it possible to rethink solutions from the past and trying to go new ways, namely the breakthrough 2007 in *neuronal networks* cutesy of G. Hinton [59]. As a result, fields like *deep machine learning*, *big data analytics* and most recently *data mining*, were gaining a wide range of attention. In almost any industry a greater amount of resources is invested in these areas [60].

The initial research motivation can be seen as a counter-movement away from the *cloud*, starting to focus again on privacy, the individual and it's digital alter ego.

From simple middleware-solutions, via full-fledged software-based platforms, through embedded hardware devices, a great variety of approaches were starting to appear in the mid 2000s until this day. A side effect was, that over time various research teams and projects have invented and coined different terms, all referring to the same concept. The following list shows some examples *(alphabetical order)*:

- Databox
- Identity Manager
- Personal …
  - Agent
  - Container
  - Data Store/Service/Stream (PDS)
  - Data Vault
  - Information Hub
  - Information Management System (PIMS)
- Vendor Relationship Management (VRM)

One of the first research projects is *ProjectVRM*, which originated from *Berkman Center for Internet & Society* at *Harvard University*. As it's name implies, it was inspired by the idea of turning the concepts of a *Customer Relationship Management* (CRM) upside down. This puts the vendor's customers back in charge of their data priorly managed by the vendors. It also solves the problem of unintended data redundancy. Over time the project has growing to the largest and most influential in this research field. It transformed into an umbrella and hub for all kinds of projects and research related to that topic [61], whether it's frameworks or standards, services offer-

ing e.g. privacy protection, reference implementations, applications, software or hardware components. *VRM* became more and more a synonym for a set of principles [62], including for example *"Customers must have control of data they generate and gather. [They] must be able to assert their own terms of engagement."* These principles can be found in various ways across a lot of research done within this area.

Another research that is worth mentioning, because of the foundational work it has been done, is the european funded project called *Trusted Architecture for Securely Shared Service* (TAS3). The project led to a open source reference implementation called *ZXID*.[11] The major goal was, to develop an architecture, that takes all involved parties into account, whether it's commercial businesses (vendors) or it's users (customers), in order to fit into more sophisticated and dynamic processes, but at the same time demanding a high level of user-centric security facilitate i.a. by a developed policy framework. Due to these requirements the architecture ended up being rather complex [63]. *ZXID* as it's implementation incorporates several standards like SAML 2.0[12] and XACML,[13] has only three third-party dependencies which are *OpenSSL*, *cURL (libcurl)* and *zlib* and as of now it supports Java, PHP and Perl. The project lasted for a period of 4 years, but after it ended in 2011, the research work has pursued i.a. by the *Liberty Alliance Project*, which is now part of the *Kantara Initiative* [64], including all documents and results. These results were taken up occasionally, recently from the IEEE [65].

A research project, which is probably the closest to what this document aims

[11] more information on the project, the code and the author, Sampo Kellomäki, can be found under *zxid.org*

[12] Security Assertion Markup Language 2.0

[13] eXtensible Access Control Markup Language

to create, bears the name *openPDS* [66] and is done by *Humans Dynamics Lab* [67], which is part of *MIT Media Laboratories*. Despite the usual concepts of a *PDS*, it introduces multi-platform components and user interfaces including a mobile devices and separating the persistence layer physically at the same time. This facilitates administrative tasks regardless of the data subject's position and time. Moreover, with their idea of *SafeAnswers* [68], the team even goes a step further. The concept behind that, is based around *remote code execution*, briefly described in one of the user stories during the first chapter. It abstracts the concept of a data request to a more human-understandable level, a simple question. This question consists of two representation: (A) a short explanation of what the data consumer wants to know and which data might be involved and thus what information a data consumer actually will receive, instead of raw data the consumer could then use for all kinds of purposes e.g. data aggregation or mining. Aside from that, the request payload also includes (B) a code-based representation, which gets executed in a sandbox on the data subjects's *PDS* system with the necessary data as arguments. The resulting output is answer and response all in once.

Aside from all the research projects done within the scientific context, applications with a commercial interest were starting to occur in a variety of sectors, too. Microsoft's HealthVault [69], for example, which aims to replace all the paper-based patient file and combine them in one digital version. This results in a patient-centered medical data and documents archive, helping doctors to make the most accurate decisions on medical treatment.

Meeco [70] [71], based on the MyData-Project [whitepaper_2014_mydata-

a-nordic-model-for-human-centered-personal-data-management-and-processing], which essentially just cuts out the advertisement service provider as a middle man inherits that role by itself. The platform does provide the data subjects with more control over what information they reveal, but it doesn't go the so eagerly demanded next step, which would means real uncoupling from the advertisement market and finding a suitable business model that focuses on the data subject, instead of surrounding them with just another walled garden.

A recently announced project, sponsored by Germany's *Federal Ministry of Education and Research*, but developed and maintained primarily by *Fraunhofer-Gesellschaft* in cooperation with several private companies like *PricewaterhouseCoopers AG*, *Volkswagen AG*, *thyssenkrupp AG* or *REWE Systems GmbH*, is the so called *Industrial Data Space* [72]. The project unifies both, research and commercial interests and runs over time period of three years until the third quarter of 2018. It aims to *"[…] to facilitate the secure exchange and easy linkage of data in business ecosystems"*, where at the same time *"[…] ensuring digital sovereignty of data owners"* [73]. It will be interesting to see how these two, yet rather distinct objectives, will come together in the future. Based on the white paper, the project's focus mainly seems to lie in enabling and standardizing the way companies collect, exchange and aggregate data with each other across process chains to ensure high interoperability and accessibility.

Hereafter a selective list can be found of further research projects, work and commercial products regarding the issue around *personal data*:

Research

- 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 of the *Solid framework*[14])
- Polis (greek research project from 2008) [http://polis.ee.duth.gr/Polis/index.php]

Organisations

- Open Identity Exchange [http://openidentityexchange.org/resources/white-papers/]
- Qiy Foundation [https://www.qiyfoundation.org/]

Commercial Products

- MyData [https://mydatafi.wordpress.com/]
- RESPECT network [https://www.respectnetwork.com/]
- aWise AEGIS [http://www.ewise.com/aegis]

2.5 Standards and Specifications

The overall attempt is to involve as much standards as possible, because it increases the chances of interoperability and thereby it lowers the effort, that might be needed, in order to integrate with third parties or other APIs. Hereinafter, some of these possible technologies will be touched on just briefly, why they might be a reasonable choice and what purposes they might going

[14]https://github.com/solid/solid

to service.

HTTP(S) [74], well known as the stateless *"transport layer"* for the *World Wide Web*, is most likely going to fulfill the same purpose in the context of this work, because it implements a server-client pattern in it very core. Whether internal components (local or as part of a distributed system) talk to each other or data consumers interact with the system, this protocol transfers the data hat need to be exchanged. Features introduced with Version 2 [75] of the protocol are yet to be known of their relevance of use cases within this project. The *Transport Layer Security* [76] embedded in the protocol provides encryption during transfer, which reduces the vulnerability to *man-in-the-middle* attacks and thus ensures data integrity. Due to it's asymmetrical cryptographic concepts used to establish a connection, *TLS* also allows to verify the integrity of the entity on the the connection's counterside, and, depending on the integration, it could even used for authentication. *Websockes* [77] might also be a possibility to communicate between components or even with external parties, which has the advantage of high efficient ongoing bidirectional connections using for real-time data exchange or remotely pending process responses, while at the same time avoiding HTTP's long-polling abilities.

JSON[15] is an alternative data serialization format to XML, heavily used in web contexts to transfer data via *HTTP*, whose syntax is inspired by the JavaScript object-literal notation.

The open standard **OAuth** defines a process flow for authorizing third parties to access externally hosted resources, such as the user's profile image

[15]The JavaScript Object Notation (JSON) Data Interchange Format; ECMA Standard [78] and Internet Engineering Task Force RFC 7159 [79]

from *facebook*. The authorisation validation is done with the help of a previously generated token. However generating and supplying such a token can be initiated in a variety of ways depending on the situation, e.g. with the user entering her credentials (`grant_type=authorization_code`). This design mistakenly [80] lead to *OAuth* integrations with the intention to provide an authentication service whether as an alternative or as an addition to existing in-house solution. Therewith the application authors pass the responsibility on to the OAuth-supporting data providers. While *version 1.0a* [81], seen as a protocol, provides integrity for transferred data by using signatures and confidentiality by encrypting data ahead of transfer. Whereas *version 2.0* [82], labeled as a framework, just requires *TLS*. It also includes certain process flows for specific platforms, such as *"web applications, desktop applications, mobile phones, and living room devices"* [83].

With **OpenID** on the other side, the authenticity of a requesting user gets verified, which is by design. An in-depth description of the whole process can be found in the protocol's same-titled open standard. With decentralisation kept in mind, the protocols's nature encourages to design a distributed application architecture, similar to the idea behind *microservices*, but without owning all services involved, *decentralized authentication as a service* so to speak. An application owner doesn't have to write or implement it's own user management system, instead it is sufficient to just integrate these parts from the standard need to support signing in with *OpenID*. Equally the user is not required to register a new account whenever it is necessary, instead she can use her *OpenID*, already created by another identity provider, to authenticate with the application. The extension *OpenID Attribute Exchange* allows to import additional profile data. *OpenID Connect* [84] is the third

iteration of the OpenID technology *Connect* is to OpenID what *facebook connect* is to *facebook*, except for the additional authentication layer, which is build upon *OAuth2.0* and therefore enables, aside from authorisation mechanisms, third parties to authenticate an OpenID-user and makes certain data available about that account via REST interface.

If it's necessary for certain components, as part of a distributed software, to make them stateless, apart from changing the architecture so that the state at that point is not needed anymore, the only other option would be to carry the state along (TODO: or "passing the state around"). This is a common use case for a **JSON Web Token** *(JWT)* [85]. A *JWT*, as it's name implies, is syntactically speaking formatted as *JSON*, but URI-safe into *Base64* encoded, before it gets transferred. The token itself holds the state. Here is where the use of *HTTP* comes in handy, because the token can be stored within the HTTP header and therefore can be passed through all communication points, where then certain data could be readout and therewith get verified. Such a token typically consists of three parts: information about itself, a payload, which can be arbitrary data such as user or state information, and a signature; all separated with a period. Additional standards define encryption *(JWE*[16]*)* to ensure confidentiality and signatures *(JWS*[17]*)* to preserve integrity of it's contents. Using a *JWT* for authentication purposes is described as *stateless authentication*, because the verifying entity doesn't need to be aware of session IDs nor any information about a state. So instead of the backend interface being constrained to check a state (`isLoggedIn(sessionId)` or `isAuthorized(sessionId)`) on every

[16]JSON Web Encryption, Internet Engineering Task Force RFC 7516 [86]
[17]JSON Web Signature, Internet Engineering Task Force RFC 7515 [87]

incoming request in order to verify permissions, it just needs

When transferring data over a potential non-private channel several properties might be desired, which eventually provide an overall trust to that data. One important aspect might me, that no one else expect sender and receiver are able to know and see what the actual data is. To achieve this, **Symmetrical Cryptography** is used for. It states that the sender encrypts the data with the help of a key and the receiver decrypts that data also with that key. This is, sender and receiver, both need to know that one key, but everyone else should not . To agree on a key without compromising the key during that process, both entities either change the medium (e.g meet physically and exchange) or have to use a procedure, in which at any point in time the entire key is not exposed to others then sender and receiver. This procedure is called **Diffie-Hellman-Key-Exchange** [88] and is based on rules for modulo operations when prime numbers are involved. It is designed with the goal to agree on a *secret* while at the same time using a non-private channel. The data exchanged during the process alone can't be used to deduce the secret. Such behaviour is similar to the concepts of **Asymmetrical Cryptography** *(or public-key cryptography)* [89], which is underpinned by a *key-pair*; one part is *public* and the other part is *private*. It depends on which of the both parts is used to *encrypt* the data, then the other part is used for *decryption*. Combining this approach with the idea of digital signatures (encrypted fingerprints of the data), then provides integrity and authentication.

REST(ful)[18] is a common set of principles to design web resources commu-

[18] *Representational State Transfer*, introduces by Roy Fielding in his doctoral dissertation [90]

nication, primarily server-client relations, in a more generic and thereby interoperable way. Aside from hierarchically structured URIs, which reflect semantic meanings, it involves a group of rudimentary vocabulary[19] to provide basic Create-Read-Update-Delete operations across distributed systems. The entire request need to contain everything that is required to get proceeded, e.g. state data and possibly authentication. These operation normally wont get applied directly to the responsible component. Instead the whole system (or certain services) exposes a restful API, with which a third party can then interact.

The *QL* in **GraphQL** [92] stands for *query language*. It's goal is to abstract multiple data sources in order to unify them under one API and make all containing data queryable, including all relating data points. The returned data, emitted in JSON syntax, can exhibit graph-like structures, meaning multiple data points, that might be somehow related to each other, or in other words: indirectly "linked" through each other. These, naturally deep-leveled structures, can be described by the syntax of the query language.

The term **Semantic Web** bundles a conglomerate of standards addressing syntax, schemas, assess control and integration around the idea of *web of data* to *"allow data being shared and reused across"* [web_2016_w3c_semantic-web-activity] or within several scopes and contexts. Alongside several others, the following three standards have a certain relevance to that concept. RDF[20] basically defines the syntax. OWL[21] provides the guidelines on how the semantics and schemas should be defined and with SPARQL [95], the query language for the RDF format, the data can be retrieved. A picture

[19]knows as HTTP Methods or Verbs [91] (e.g. GET, OPTIONS, PUT, DELETE)
[20]Resource Description Framework [93]
[21]Web Ontology Language [94]

emerges in which the web is used as a database, queried by URIs with a query language. An example would be a person's email address, which is available under a specific domain (preferable owned by that person) - or to be more precise, an URI *(WebID) [96]* - and provided in a certain syntax *(RDF)* and tagged with the semantic *(OWL)* of a email address; all embedded in a valid html page. This information can be queried *(SPARQL)*, which requires at least the URI, working as a unique identifier. While defining the standards, an importancy was to define a syntax which is also valid markup, in order to maintain a single source of trough and save redundant work. Related to this topic is the work on a specification called **Solid**.[22] Based on the *Linked Data* principles, that are facilitated through the standards just mentioned and the *WebAccessControl* [98] system, the project focuses on decentralization and personal data. A reference implementation called *databox* [99] combines all these technologies and is build on top.

The concept of application (or software) **container** is about encapsulating runtime environments by introducing an additional layer of abstraction. A container bundles just the software dependencies (e.g. binaries) that are absolutely necessary so that the enclosed program is able to run properly. The actual container separation is done, aside from others, with the help of two features provided by the Linux kernel. *Cgroups*,[23] which define or restrict how much of the existing resources a group of processes (e.g. CPU, memory or network) can use. Whereas *namespaces* [101] define or restrict what parts of the system can be accessed or seen by a process (e.g. filesystem, user, other processes). The idea of encapsulating programs from the operating

[22] social linked data [97]
[23] control groups [100]

system-level is not new, Technologies, such as *libvirt*, *systemd-nspawn*, *jails*, or *hypervisors* (e.g. VMware, KVM, virtualbox) have been used for years, but were usually too cumbersome and never reached a great level of convenience, so that only people with a certain expertise were able to handle systems build upon virtualization, but people with other backgrounds couldn't and weren't that much interested. Until *Docker* and *rkt* emerged. After some years of separated work, both authors, and others, recently joined forces in the *Open Container Initiative* [102], which aims to harmonize the diverged landscape and start building common ground to ensure a higher interoperability, and that in turn is requisite for orchestration. It also marks the initial draft of the specifications for runtime [103] and image [104] definition, on which the work is still ongoing. This concept of *containerization* also inherits the a ability known from *emulation*, because it allows a certain set of software to run on a system that otherwise is not supported, e.g. mobile devices. It only requires the runtime to be working.

3

Core Principles

Right from the start a set of principles have build the cornerstones and orientation marks of the idea behind the *PDaaS*. Those, who meant to be reflected also by the arising *Open Specification*, will be explained further within the following sections.

3.1 Data Ownership

Depending on the standpoint, the question about ownership of certain data might not that trivial to answer. As stated in the previous section, owner-

ship requires a certain amount of originality to become intellectual property, which is not the case for personal data - at least for all the non-creative content. Thus there is no legal ground for an individual to license those data, that obviously belongs to her. Switching the perspective from the *data subject* to the data *consumer*; for them, several laws exist addressing conditions and rules regarding data acquisition, processing and usage. Leaving aside the absence of any legislation regarding data ownership, it can not be denied, that is seems unnatural not being the owner of all the data that reflects her identity and her as an individual. So instead of defining those rules meant to protect data subjects, but demanding data consumers to comply with, the proposal here is to put the entity, to whom the data is related to, in control of defining, who can access her data and what accessor is allowed to do with it. This would make the *data subject*, per definition and effectively to the owner of those data. Although, it is to be noted, that the legal rulebook for data consumers mentioned before, remains a highly important, since this project is not able to cover every use case, that might occur.

Promoted from the data subject to the data owner, thus being the center of the *PDaaS*, the operator gains abilities to have as much control as possible over all the data related to her, to determine in a very precise way what data of hers can be accessed by third parties at any point in time and to literally carry all her personal data with her.

3.2 Identity Verification

When an instance of this system is going to be the digital counterpart of an individuals identity or it's *personal agent* [105], then everyone who relies on the information that agent is providing, must also be able to trust the source from where that data is coming from and vice versa; the *operator* must be able to verify the authenticity of the requesting source, too; regardless if it's the initial *permission request* or further *access attempts*. Based on these mechanisms, the system can also provide an authentication services to all sorts of generic or restricted platforms for the associated identity, including second factor abilities.

3.3 Reliable Data

Being able to verify the authenticity of a communication partner means only to be half-way through. Data consumers also need to trust the data itself, which is attributed to the following properties.

(A) *integrity* - which means the recipient can verify, that the data, sent to her, is still the same, or if someone has tampered with the obtained data. (B) *authenticity* - it is somehow ensured, or the recipient must be certain, that the received data belongs to the individual from whom the data comes from. A negative result of that check should not cause a termination of the process, but instead should warn the recipient about the lack of authenticity, so that she, herself, can decide if and how to proceed.

3.4 Authorisation

Controlling it's own data might probably be the most important ability of such a system, because the data owner gets enabled to grant permission to any entity who want to obtain certain information about her in a semi-automated way. She can authorise as precise as desired how long and what data (sets, points or fields) is accessible by a single entity. Thereby, the data owner is able to change the *access permissions* for any entity at any point in time, for example motivated by a noticed incident.

3.5 Supervised Data Access

Rules and constraints might be one way to handle *personal data* demands of *third parties*. But this plain *query and response data* approach could be replaced by a more supervised concept, that prevents data from leaving the system. It allows to execute a small program within a locally defined environment, computing only a fraction of a larger computation that was initiated by the *data consumer* beforehand; similar to a distributed Map-Reduce concept [106]. The opposite approach, to provide some software to the *data consumer* that is necessary to access the contents of a response or provides a runtime environment querying the system by itself, would be conceivable, too. In general, it is not very likely that *data consumers*, who already got granted certain access, would renounce their privileges. Thus it is vital that the *data owner* is the one who is able of cancel the *access permissions* or applying appropriate changes. Supervising methods provide an appropriate ways to make data available to those who are eager to consume them.

3.6 Containerization

Abstracting an operating system by moving the bare minimum of required parts into a virtualization results into an environment that can be, depending on the configuration, fully encapsulate it's internals from the host environment. This approach yields to some valuable features. Such as:

(A) Effortless portability, which reduces the requirements on environment and hardware to a minimum.

(B) Thereby gaining higher flexibility in placing components, through which advantages can be made out of other devices characteristics. while not necessarily increasing the overall complexity of the system

(C) Isolation and reduction of shared spaces and scopes, which for example can prevent side effects.

    All these in conjunction lead also to an overall security improvement or at least it enables new patterns to improve such aspects. Furthermore, it allows to suit more versatile and diverse scenarios, like storing data about a using data, providing sensitive profile data or getting used as a patient file. The convenience of a precise resource assignment might also become relevant for case where device's hardware specification might be somewhat low. Building a system upon a container-based philosophy and enclosing components in their own environment brings a variety of design and architectural possibilities without the necessity of increasing the overall system complexity.

3.7 Open Development

When developing an *Open Specification* it only comes natural to build upon open technologies, which are understand as *open standards* and *open source*; *open* in the sense of *unrestricted accessible by everybody* and not to be confused with free - as in *freedom* - software. Advocating such a philosophy permits not only to develop implementations in a collaborative way, but enables

also to work fully transparent on the specification itself. Such an open environment makes it possible for anyone who is interested, to participate or even to contribute to the project. Thus, to lower the barrier, usable and meaningful documentation is vital. Such an openness ensures the possibility of looking into the source code and getting a picture of what the program actually does and how it works. Thus, source code reviews become possible as well. Those might reveal certain security flaws, which then are able to get fixed very quickly. Furthermore, this approach allows data subjects to setup their own infrastructure and host such a system, which gains even more control over the data and increases the level of trust, instead of using a *SaaS*[1] solution that is host by another provider. It also encourages any kind of adjustments or customization to the system in order to serve the own's needs. Enabling an open development allows users and contributors working together and thus improve the outcome in a variety of ways.

[1] Software as a Service

4

Requirements

Derived from the Core Principles, the subsequent requirements shall be served as a list of features on the one hand, to get an idea about how the open specification and thus the resulting software might look like, and to give an overview about priorities (can/could, may/might, should, must/have to) on the other hand. Other chapters may contain specific references to the requirements listed below.

4.0.0.1 Architecture/Design:

S.A.01 **- Accessibility & Compatibility**

Since the internet is one of the most widely used infrastructure for data transfer and communication, it is assumed that all common platforms support underlying technologies, such as HTTP and TLS. Thus the emerging system should implement a web service, who provides supervised access to personal data.

S.A.02 **- Portability**

All major components should be designed and communicate between each other in a way to be able to get relocated while the system has to remain fully functional. It has to be possible to build a distributed system, that may require to place certain components into different environments/devices.

S.A.03 **- Roles**

The system has to define two types of roles. The first one is the operator, who is in control of the system and, depending on the architecture, must be at least on individual but can be more. The operator takes care of all the data that then get's provided and decides about which third party get's access to what data. The second type are the consumers. These are external third parties that desire certain data about or from the operator. (see Terminologies)

S.A.04 **- Authenticity**

Since they have to rely on the data, both entities - everyone who belongs to one of the *roles* - have to be able to ensure the authenticity of their identity and the data they are sending to the opponent. It should be possible to

opt out to that level of reliability, if is not necessary. Whereas if one of the parties demanding the other one of providing such level, but the other doesn't, then the access attempt has to fail.

4.0.0.2 Persistence:

***S.P.01* - Data Outflow**

Data may only leave the system if it's absolutely necessary and no other option exists to preserve the goal of that process. But if data still has to get transferred, no other than the data consumer must be able to access the data. Confidentiality has to be preserved at all cost.

***S.P.02* - Data Relationship**

Data structures and data models must show high flexibility and may not consist of strong relations and serration.

***S.P.03* - Schema and Structure**

The *Operator* can create new data types (based on a schema) in order to extend the capabilities of the data API. Structures and schemas can change over time (S.P.04). Every data set and data point has to relate to a corresponding and existing type, whether it's a simple type (string, integer, boolean, etc.) or a structured composition based on a schema.

***S.P.04* - Write**

Primarily the operator is the only one who has the permissions to add, change or remove data. This is done either by using the appropriate forms provided by visual user interface or import mechanisms. The latter could be enabled through (A) support for file upload containing supported formats, (B) data

API restricted to the operator or (C) defining an external source reachable via http (e.g. *RESTful URI*) in order to (semi-)automate additional an ongoing data import from multiple data sources (e.g. IoT, browser plugin). Additionally, it might be possible in the future to allow *data consumers* letting some data to flow back into the operator's system, after she is certain about it's validity and usefulness.

4.0.0.3 Interfaces:

***S.I.01* - Documentation**

The interfaces of all components have to be documented; in a way that the components themselves can be replaced without any impact to the rest of the system. This also involves comprehensive information on how to communicate and what endpoints are provided, including required arguments and result structure.

***S.I.02* - External Data Query**

Data consumer can request a schema, in order to know how the response data will actually look like, since certain parts of the data structure might change over time (see S.P.03, S.P.04). After checking if the access request is permitted, the system first parses and validates the query and eventually proceeds to actually execute the included query. When querying data from the system, the *data consumer* might be required tp provide a schema, which should force him to be as precise as possible about what data is exactly needed. In addition to that, the consuming entity must provide some *meaningful* text, describing the purpose of the requested data. He should not be allowed to place wildcard selectors for data points in the query. Instead

he must always define a more specific filter or a maximum number of items, if the query retrieves more then one element.

***S.I.03* - Formats**

When components communicating between each other or interactions with the system from the outside take place, all data send back and forth should be serialized/structured in a JSON or JSON-like structure.

4.0.0.4 Visual User Interface:

***P.VIU.01* - Responsive user interface**

The visual user interface has to be responsive to the available space, because of the diversity of screen sizes nowadays.

***P.VIU.02* - Platform support**

The user interface must be at least implemented based on web technologies, that is provided by a server and is thus available on any platform that comes with a modern browser. To enable additional features and behavior, at least for mobile devices it is recommended to build a user interface upon native supported technologies, such as *Swift* and *Java*. The operator would benefit from capabilities such as *push notifications* and storing data on that device.

***P.VIU.03* - Access Profiles**

The operator should be capable of filtering, sorting and searching through the list of *access profiles*; for a better administration experience and to easily find certain entries while the overall amount increases over time.

***P.VIU.04* - Access History** The operator must be provided with a list of all past permission requests and data accesses, in order to monitor who

is accessing what data and when, and thus being capable of evaluating and eventually stopping certain access and data usage. This tool should have filter, search and sort capabilities. It is build upon and therefore requires the access logging functionality.

4.0.0.5 Interactions:

***P.I.01* - Effort**

Common interactions processes, like changing *profile data*, importing data sets or manage *permission request* have to require as little effort as possible. This means short UI response time on the one hand and as less single input and interaction steps as possible to complete a task. Given these circumstances, the *permission request review* and *access profile creation* might become a special challenge.

***P.I.02* - Design**

The visual user interface must be designed and structured in such a way that is is highly intuitive for the user to operate. Thus, it is important e.g. to use meaningful icons and appropriate labels. It also means a flat and not crammed menu navigation. Context related interaction elements should be positioned within the area designated for that context. TODO: maybe emphasize more UI aspects (or not)

***P.I.03* - Notifications**

The user should be notified about every interaction with the *PDaaS* originated by a third party immediately after it's occurrence, but she must get notified at least about every *permission request*. This behaviour should be

configurable; depending on the *permission type* and on every *access profile*. Regardless of the configuration the notifications themselves must show up and pending user interactions must be indicated in the user interface.

P.I.04 - Permission Request & Review

A process involving data transaction must always be initiated by the data subjects. So before a *data consumer* is able to access data, first the *operator* need to *invite* him and tell him whereto address his requests. This has to be done by sending him a URI leading to an endpoint, that needs to be unique among all *data consumers* interacting with the same instance of the system. When a *data consumer* makes the first attempt to connect to the system, it must be a well formed *permission request*, which has to include information about the *consumer*, what data he wants to get access to, for what purpose and how log or how often the data need to be requested. The operator then reviews these information and creates an access profile based on that information. A key configuration in such a profile has to be what defines when this permission expires. The operator should be able to decide between three *permission types*:

- *one-time-only*
- *expires-on-date*
- *until-further-notice* After creating the profile, a response must be send to the *data consumer*, which should contain the review result and permission type set by the operator.

P.I.05 - Templating

The operator should be able to create templates for *access profiles* nad *permission rules* in order to (A) apply a set of configuration in advance before

the *permission request* arrives and

(B) reduce recurring redundant configurations.

4.0.0.6 Behaviour:

***P.B.01* - Access Logging**

All interactions and changes in the persistence layer should be logged. At least all data request must be logged. Such log is the foundation of the *access history*, with this the user is able to keep track of and look up past accesses.

***P.B.02* - Real time**

Real time communication might be essential for time-critical data transaction. Hence at least one user interfaces should be connected to the server through an ongoing connection to enable real time support (example scenario: permission request got reviewed on mobile device, but notification indicator reflects "still pending"). But if just one client is associated to the system, real time (in the sense of keeping UI state up to date) would not be necessary. (see P.VIU.02)

5

Design Discussion

The following chapter documents the processes of some design decision makings, examines possible issues emerging alongside and discuses different solutions obtained from several perspectives in order to evaluate their advantages and disadvantages. Probably not every issue will get it's
deserved room, but major aspects will be addressed. In short, the majority of the project's conceptual work is done below.

Eevery subchapter includes at the end a section containing a summary of conclusions, which are based on the prior discussions about the related topic.

5.1 Architecture

Within this sections questions such as

- how can a communication process with a third party be modeled and what technologies can be used to

(A) trust the system and
(B) trust the communication partner

- how many and what kind of authentication mechanisms are required?
- how can data be provided to a a data consumer without the data ever leaving the system?
- where are reasonable places to locate the storage that holds the operators's personal data

5.1.1 Authentication

First of all, the system has to support two roles[1]. Any entity can be assigned to either one of them, hence entities that are trying to authenticate to the system might have different intentions. The *operator* for example wants to review *permission requests* in real time, so accessing the system from different devices is a common scenario. When inheriting the *operator role* an entity gains further capabilities to interact with the system, such as data manipulation. Whereas a *data consumer* always uses just one origin and processes requests sequentially. Those very distinct groups of scenarios would make it possible to apply different authentication mechanisms that do not

[1] %7B#sa03

necessarily have a lot in common.

With respect to the requirements (S.A.01), the most appropriate way to communicate with the *PDaaS* over the internet would be by using *HTTP*. Furthermore, to preserve confidentiality on every in- and outgoing data (S.P.01) the most convenient solution is to use *HTTP* on top of *TLS*. *TLS* relies i.a. on asymmetric cryptography. During the connection establishment the initial handshake requires a certificate, issued and signed by a CA [ˆabbr_ca], which has to be provided by the server. This ensures at the same time a seasonable level of identity authentication, almost effortless. If the certificate is not installed, it can be installed manually on the client. If the certificate is not trusted (e.g. it is self-signed), it can either be ignored or the process fails to establish a connection, depending on the server configurations. The identity verification in TLS works in both directions, which means not only the client has to verify the server's identity by checking the certificate. If the server insists on, the client has to provide a certificate as well, which then the server tries to verify. Only if the outcome is positive, the connection establishing succeeds. According to the specification [107] it is still optional though.

HTTP as a comprehensive and flexible protocol enables to use several technologies for server-client authentication purposes. Within the scope of this work, those technologies are categorized in the following types (TODO: maybe find other labels): (A) stateful and (B) stateless authentication. The first one (A) includes vor example *Basic access Authentication* (or *Basic Auth*) and authentication based on *Cookies*. Whereas the *two-way authentication* in TLS mentioned above and authentication based on web-token are

associated with the latter (B). *Basic Auth* is natively provided by the *http-agent* and requires in it's original form (*user:password*) some sort of state on the server; at least when the system has to provide multitenancy. If instead just a general access restriction for certain requests would suffice, no state is required. One of the most common implementations of user-specific states is a *session* on the server, that contains one or more values representing the state and a unique identifier, by which an entity can be associated with. A client has to provide that session ID in order to get provided with all the session-related data hold by the server. This is typically done in a HTTP header, whether as *Basic Auth* value, a *Cookie*, which is domain-specific, or in some other custom header. Since the *two-way authentication* (or *mutual authentication* [108]) is done based on files containing keys and certificates, which are typically not very fluctuant in it's contents or state, this procedure is categorized as stateless. Order or origin of incoming requests have no impact on the result of the actual authentication process. The same applies to TLS features such as *Session [ID, Ticket] Resumption* [109], thus they are left aside, because they serve the sole purpose of performance optimization. Similar to the *Session Ticket Resumption* [110] a web token, namely the JSON Web Token, also moves the state towards the client, but that's about all they have in common. A *JWT* carries everything with it that's worth knowing, including possible states, and if necessary the token is symmetrical encrypted by the server. This is, only the server is able to obtain data from it and reacting accordingly.

Keeping track of a state (or multiple states) on the server and keeping data that is involved

synchronized between server and client is expensive and by fare trivial. Ex-

pensive in the sense of additional resources a server would require to remember all the data for those states, that otherwise won't be needed. And it's not trivial, because this pattern requires the server to be aware of all current states (sessions) and has to have them accessible at all time. This also means, that the contents responses for certain requests might depend on preceding requests and their incoming order. Furthermore those session data has to be safely stored from time to time. Otherwise if the server fails to run at some point, data only existing in the memory would be gone without any possibility to get recovered. To stateless authentications non of those aspects apply. Certificates and keys as well as web tokens are both carry the information that might be necessary with them. Thus, considering those disadvantages, *public key cryptography* and web tokens are the preferred technologies for all authentication processes.

Because of it's simplicity the concept of web tokens are fairly straightforward to implement into the *PDaaS*. And since web tokens ensure integrity and perhaps also the confidentiality only of their own carriage and not the entire HTTP payload, both requirements need to be addressed separately. Serving HTTP over *TLS* solves this issue, but using TLS or *asymmetric cryptography* properly - place value on integrity, confidentiality, authenticity - requires additional infrastructure. Such an infrastructure is known as *Public Key Infrastructure* (or *PKI*) [111]. It manages and provides public keys in a directory, including related information to the entities those certificates belong to. A Certificate Authority (or *CA*), as part of that infrastructure, issues, maintains and revokes digital certificates. The infrastructure that is needed to provide secure HTTP connections for the internet can be seen as

a large, if not the largest, public *PKI*. It is based on the widely used IETF[2] standard *X.509* [112]. For connections that use a web token, it is sufficient to rely on the public PKI that drives *HTTP* over *TLS*, because unlike the *two-way authentication*, authentication is provided by the token itself. The situation is different, if instead *two-way authentication* is used. For this the system has to provide it's own *PKI* including a Certificate Authority that issues certificates for *data consumers*.

- maybe place "eperso + de-mail" here

An advantage of token-based authentication over TLS-based *mutual authentication* is that the token can be used on multiple clients at the same time or the account a token is associated with can actually have more than one token. Whereas during the asymmetric cryptography-based *mutual authentication* the client's private key is required. Such key should not leave it's current place in order to prevent exposure, which implies any action of duplication.

If a public-key-based connection, performing a *mutual authentication*, establishes successfully, it implies that the requester's identity is valid and the integrity of the containing data is given. Whereas on a token-based authentication every incoming request has to carry the token so that the system can verify and associate the request with an account. Data it not automatically encrypted and thus integrity is not preserved.

- CA might be part of a chain of trusted CAs

To hardening an authentication procedure often one ore more factors are added. This makes the procedure more complex and thus increases the ef-

[2]Internet Engineering Task Force; non-profit organisation that develops and releases standards mainly related to the Internet protocol suite

fort that's needed for succeeding attacks. Using multi-factor authentication is generally valued and will be briefly noted as an optional security enhancement for the *operator role*. However detailed discussions regarding this topic are left to follow-up work on the specification.

An endpoints is defined as the part of the URI that is unique to every *data consumer*, or to be more precise, unique to every *access profile*.

Since there are no time constrains when it comes to communication with a payload containing personal data, parameters for encryption procedures can chosen as costly as the system resources allow them to be, thus the level of security can be increased.

5.1.2 ePerso, E-Post/de-mail as part of a PKI solution?___

In the past years different countries around the world recently started to introduced *information technology* to the day-to-day processes, interactions and communications between public services and their citizens, for example changing residence information or filing tax report, which is summarized under the term *E-government*.[3] One of those developments is the so called *electronic ID card*, hereinafter called *eID card*. Equipped with storage, logic and interfaces for wireless communication, those *eID cards* can be used to store certain information and digital keys or to authenticate the owner electronically to a third party without being physically present. Such an *eID card* was also introduced in Germany in 2010. The so called *nPA*[4] was an important step towards an operational *e-government*.

Aside from minor flaws [113] and disadvantages [114] an *eID card* can have, the question here is, how can this technology be usefully integrated in this

[3]Electronic government

[4]in german so called *elektronische Personalausweis (nPA)*

project and does it even makes sense. As an official document the card has one major advantage over self-configured authentication mechanisms like password, fingerprints or second factor tokens. It is *signed* by design, meaning by creating this document and handing it over to the related citizen, the third party - in this case the government - has verified the authenticity of that individual.

As a result, several ideas can be proposed:

(A) authenticate with the *eID card* to the management UI of the *PDaaS*

(B) authorize/approve *access requests* or *data access* attempts

(C) sign the responding data, in order to not only preserve data integrity but also to prove the authenticity of the data.

With regards to (A), partially depending on the *eID card*'s implementation, but in general this use case would make sense. When considering the german implementation (nPA), accessing the management UI via desktop would require just a card reader - preferable with a hardware keypad attached. Accessing the UI via mobile device could be achieved with the card's RFID-capabilities, as long as the used device is able to communicate with the RFID-chip. Both cases need the *nPA* to have enabled the *eID* feature. If a service wants to provide *nPA*-based online authentication (*eID-Service*), which is defined as a non-sovereign ("nicht hoheitlich") feature, it has to comply with several requirements [115] starting with making an application in order to get permission for sending a certificate signing request to a BerCA.[5] This request is originated from an *eID-Server* [116] to sign a public key generated on a dedicated and certified hardware, which is also re-

[5]Berechtigungszertifikate-Anbieter

quired through the officials. This key pair - re-generated and re-signed every three days - is needed to establish a connection with the *nPA*, which then is used to authenticated the owner of that *ID card*. The described appears to be highly expensive (effort, hardware costs), especially because every single operator needs to go through the whole process in order to provide this authentication method; not mentioning the uncertainty of the official's decision about the motion filing. Another approach would be to integrate an external authentication provider supporting the *nPA*, which would not only add an additional dependency, but could also weaken the system.

(A) and (B) are fairly similar, insofar as they would use the same mechanism to authenticate, but to approve different actions.

As of (C), as mentioned above the *nPA* is able to store data required for creating a *QES*[6] (e.g. private keys and certificates). Thus this procedure could be used to sign arbitrary data, normally messages within communications like emails. But since a signing procedure involves the operators private key, every time data get accessed by a consumer, she is forced to interact withe the system. Otherwise the operators private key need to be stored somewhere within the system. No matter where, this approach would potentially expose a highly confidential part of a cryptographic procedure, which not only reduces the overall system's security level. It also makes every process this procedure is involved with vulnerable to certain attacks. Apart from that, it is highly unlikely that a *eID card* does allow to extract private keys. Thus increasing inconvenience is inevitable for such a use case.

Another technology, emerging as part of the *e-government* development, is

[6]Qualified Electronic Signatures [117]

the *DE-Mail* [118]. It's an eMail-Service that is meant to provides infrastructure and mechanisms to exchange legally binding electronic documents. One would expect public-key-based cryptography procedures all the way from sender through to the recipient (end-to-end) [119], maybe even with taking advantage of the *nPA's* capability to create *QES*. Instead, the creators of the corresponding law decided that it's sufficient prove about the documents author if the provider signs the document on the email server and that this implementation results in a legally binding document by definition of that law. This technology tries to embed a legal foundation into email-based communication, thus has no relevance to this project, other then letting a server sign outgoing data, which might be the only solution to avoid an overhead in user interaction caused by recurring events.

Conclusions: Based on the several requirements and distinct advantages of the two authentication mechanisms, it would make sense to use asymmetric cryptography in combination with *HTTPS* for the communication between the system and *data consumers*, where the system provides it's own *PKI* and a token-based authentication on top of *HTTPS* and public CAs for communication between the system and the *operator*, preferable based on *JSON Web Tokens*, because the session state is preserved within the token rather then having the system itself keeping track of it.

5.2 Access Management

In the previous section [Authentication] has been addressed. The Subsequent section discusses how the mentioned technologies has to be assembled

in order to meet the requirements (TODO), so that personal data can be accessed by external *data consumers*.

While in OAuth the authorisation procedure strictly involves an authentication, the previous proposed design separates authentication and authorisation from each other so they can run completely independent. Additionally this approach would require almost no effort to support the case where multiple *data consumers* access the same *endpoint*. by just disabling the client authentication for the HTTPS connection establishment.

A) just requesting and responding with pure data

B) provide executable and input schema; run on the PDaaS environment in a sandbox; return result

C) DRM for personal data: provide a piece of software to the data consumer, which does the licence checking, key obtaining and encryption of the data, he has requested, in order to work with / compute that data

5.3 Components

- which components can go where?

Webserver

- to serve UI
- relay to mobile device

UI

- data editor and importer
  - data type editor
- permission management
  - access history and access profile

Storage/Persistence

- regardless of the platform
- connector
- where to place the storage? local (e.g. mobile device) or cloud (e.g. hoster's infrastructure)
  - requires 24/7 uptime

Notification Infrastructure

- websockets for web UIs
- Google/Apple Notification server compatible connection for mobile apps

Data API

- essentially consists of two parts:
  1) checking permissions of the request
  2) persistence layer abstraction (graphql)
- for external consumers
  - incoming permission requests and data access attempts
  - outgoing data ()
- for internal clients (web UI, mobile device)

Conclusions:

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

5.4 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

5.4.1 Authenticity

Within the section about architecture[7] several options were discussed on how to preserve data integrity, referring to possible man-in-the-middle attacks and alike.

With authenticity here it is referred to the originality of the data, meaning

(A) the data really represent the entity associated to the originating system and

(B) the data is true at the time when the response leaves the originating system

requirement

- maybe go with a Signing/verifying Authority (aka CA)
  - do I trust the gov or certain companies more? Which interests do

[7] TODO

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
- Where?
  * direct at a Meldestelle
  * with the eID-Function (data check: as long as the contents of the relevant fields are equal) of the Personalausweis –> no, because every PDaaS need its own permission cert from the authorities

5.4.2 Modelling

5.4.3 Categories (or Classes)

5.4.4 Types

5.4.5 Persistence

- database requirements

5.4.6 Access & Permission

- data needs to have an expiration date

IF01 - Authorizing a data consumer to request certain data

1) operator creates a new endpoint URI (like *pdaas.datasubjectsdomain.tld/e/consumer-name*) within the *management user interface*
2) operator 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) operator 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 operator's decision (NOTE: alongside with some details? how do they look like? XXX need to be in the spec)

5.4.7 Consumption (data inflow)

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

5.4.7.1 Manually

5.4.7.2 Automatically

5.4.8 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.

5.4.9 History

- data versioning
- access logs

5.4.10 *Conclusions*

5.5 Interfaces

5.5.1 Internal

- UI for Management & Administration

5.5.2 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)

5.5.3 Authentication

- regarding oAuth as authentication: Priorly users tended to reuse their password for different account, nowadays they but also tend to get tired of creating new accounts and profiles over and over again instead

of having just one account for everything [120].

Thus the platform owners leave the responsibility of

Many websites and platforms understand those *login-with $platformName* mechanisms as an outsourced service that handles all security- and user-related tasks.

what about token stealing when using jwt?

5.5.4 *Conclusions*

6

Specification

6.1 Overview

- purpose
- architectural overview
- short description of the whole process

6.2 Components

6.2.1 Webserver

6.2.2 User Interface

6.2.3 Storage/Persistence

6.2.4 Notification Infrastructure

6.2.5 Data API

6.3 Data

6.3.1 Structure & Types

6.3.2 Read

6.3.3 Write

6.4 Protocols

6.4.1 Permission Request / Consumer Registration

6.4.2 Data Access

6.4.3 Data Management

6.5 APIs

How do the APIs involved with the protocols look like?

6.6 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/occurrence with a learning artificial intelligence
- 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`

6.6.1 Environment

6.6.2 Transport

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

6.6.3 Storage

- documents based DB instead of Relational DBS, because of structure/model flexibility
- graphql because of it's nature to abstract a storage engine, which comes in handy when the actual storage gets relocated (e.g. from a server to a mobile device)

6.6.4 Authentication

- how should consumer authenticate?

6.7 Recommendations

6.7.1 Software Dependencies

6.7.2 Host Environment(s)

7

Conclusion

7.1 Ethical & Social Impact (TODO: or "Relevance")

7.2 Business Models & Monetisation

- possible resulting direct or indirect business models
- data subject 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

- adoption rate of such technology

7.4 Solutions

7.5 Attack Scenarios

- single point of failure (data-wise),
  - but considering what data users already put into their social networks (or: thE social network: fb), they/it has already become a de facto data silo and is thus a single point of failure. If that service breaks or get down, the data from all users might be lost or worse (stolen). The aspect of data decentralisation achieved by individual data stores can be valued as positive.

7.6 Future Work

- maybe enable the tool to play the role of an own OpenID provider?
- going one step further and train machine (predictor) by our self with our own data (https://www.technologyreview.com/s/514356/stephen-wolfram-on-personal-analytics/)

7.7 Summary

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

The work will be continued.

[1] “Big data privacy international.” [Online]. Available: https://www.privacyinternational.org/node/8. [Accessed: 15-Nov-2016]

[2] 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]

[3] S. Spiekermann, *Ethical IT Innovation: A Value-Based System Design Approach*. CRC Press; Taylor & Francis Group, LLC, 2015, pp. 66–72 [Online]. Available: https://www.crcpress.com/Ethical-IT-Innovation-A-Value-Based-System-Design-Approach/Spiekermann/p/book/9781482226355

[4] B. Friedman and H. Nissenbaum, “Bias in computer systems,” *ACM Transactions on Information Systems (TOIS)*, vol. 14, no. 3, pp. 330–347, 1996 [Online]. Available: http://dl.acm.org/citation.cfm?id=230561. [Accessed: 07-Nov-2016]

[5] “Cognitive bias,” *Wikipedia*, Oct-2016. [Online]. Available: https:

//en.wikipedia.org/w/index.php?title=Cognitive_bias&oldid=742803386. [Accessed: 08-Nov-2016]

[6] R. Lemov, "Why big data is actually small, personal and very human. Aeon essays," 16-Jun-2016. [Online]. Available: https://aeon.co/essays/why-big-data-is-actually-small-personal-and-very-human. [Accessed: 17-Nov-2016]

[7] A. Dewes, "C3TV - Say hi to your new boss: How algorithms might soon control our lives." 29-Dec-2015. [Online]. Available: https://media.ccc.de/v/32c3-7482-say_hi_to_your_new_boss_how_algorithms_might_soon_control_our_lives#video&t=1538. [Accessed: 03-Nov-2016]

[8] "ProjectVRM - about. ProjectVRM," 25-Feb-2010. [Online]. Available: https://blogs.harvard.edu/vrm/about/. [Accessed: 09-Nov-2016]

[9] Tom Kirkham, Sandra Winfield, Serge Ravet, and S. Kellomaki, "The personal data store approach to personal data security," *IEEE Security & Privacy*, vol. 11, no. 5, pp. 12–19, 2013.

[10] A. Poikola, K. Kuikkaniemi, and H. Honko, "MyData – a nordic model for human-centered personal data management and processing," pp. 1–12, Jun. 2015 [Online]. Available: https://www.lvm.fi/documents/20181/859937/MyData-nordic-model/2e9b4eb0-68d7-463b-9460-821493449a63. [Accessed: 10-Nov-2016]

[11] "Meeco how it works." [Online]. Available: https://meeco.me/how-it-works.html. [Accessed: 09-Nov-2016]

[12] "Open specification of the concept called personal data as a service

(pdaas). GitHub." [Online]. Available: https://github.com/lucendio/pdaas_spec. [Accessed: 11-Nov-2016]

[13] "ProjectVRM wiki - about VRM." [Online]. Available: https://cyber.harvard.edu/projectvrm/Main_Page#About_VRM. [Accessed: 11-Nov-2016]

[14] "ProjectVRM wiki - list of personal information management systems." [Online]. Available: https://cyber.harvard.edu/projectvrm/VRM_Development_Work#Personal_Information_Management_Systems_.28PIMS.29. [Accessed: 11-Nov-2016]

[15] F. T. C. USA, "Data brokers," May 2014 [Online]. Available: https://www.ftc.gov/system/files/documents/reports/data-brokers-call-transparency-accountability-re 140527databrokerreport.pdf. [Accessed: 17-Nov-2016]

[16] J. Rose, O. Rehse, and B. Röber, "The value of our digital identity," *Boston Cons. Gr*, 2012 [Online]. Available: https://www.libertyglobal.com/PDF/public-policy/The-Value-of-Our-Digital-Identity.pdf

[17] "Outline of intellectual property," 11-Oct-2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Outline_of_intellectual_property&oldid=743830160. [Accessed: 25-Dec-2016]

[18] *General data protection regulation.* 2016, p. L 119/33 [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[19] Wikipedia, "Information privacy law," 13-Nov-2016. [Online]. Available: https://en.wikipedia.org/wiki/Information_privacy_law#United_States.

[Accessed: 20-Nov-2016]

[20] I. J. (Loeb & Loeb), "PLC - data protection in the united states: Overview," 01-Jul-2013. [Online]. Available: http://us.practicallaw.com/6-502-0467. [Accessed: 20-Nov-2016]

[21] A. Wilhelm, "White house drops 'consumer privacy bill of rights act' draft. TechCrunch," 27-Feb-2015. [Online]. Available: http://social.techcrunch.com/2015/02/27/white-house-drops-consumer-privacy-bill-of-rights-act-draft/. [Accessed: 20-Nov-2016]

[22] *Administration discussion draft: Consumer privacy bill of rights act of 2015*. 2015 [Online]. Available: https://www.whitehouse.gov/sites/default/files/omb/legislative/letters/cpbr-act-of-2015-discussion-draft.pdf

[23] *Report and order*. 2016 [Online]. Available: https://transition.fcc.gov/Daily_Releases/Daily_Business/2016/db1103/FCC-16-148A1.pdf. [Accessed: 20-Nov-2016]

[24] *Notice of proposed rulemaking*. 2016 [Online]. Available: https://apps.fcc.gov/edocs_public/attachmatch/FCC-16-39A1.pdf. [Accessed: 20-Nov-2016]

[25] "Privacy policies are mandatory by law," 23-Oct-2016. [Online]. Available: https://termsfeed.com/blog/privacy-policy-mandatory-law/. [Accessed: 20-Nov-2016]

[26] "International privacy standards," 29-Sep-2016. [Online]. Available: https://www.eff.org/issues/international-privacy-standards. [Accessed: 20-

Nov-2016]

[27] G. Rosner, "Who owns your data?" presented at the UbiComp '14, september 13 - 17 2014, seattle, wa, usa, 2014, pp. 623–628 [Online]. Available: http://dl.acm.org/citation.cfm?doid=2638728.2641679. [Accessed: 01-Dec-2016]

[28] J. Grunebaum, *Private ownership*. Routledge & Kegan Paul, 1987, p. 213.

[29] *General data protection regulation*. 2016, p. L 119/12 [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[30] *Report and order*. 2016 [Online]. Available: https://transition.fcc.gov/Daily_Releases/Daily_Business/2016/db1103/FCC-16-148A1.pdf. [Accessed: 20-Nov-2016]

[31] Facebook, "Facebooks's terms of service. Statement of rights and responsibilities," 30-Jan-2015. [Online]. Available: https://www.facebook.com/legal/terms. [Accessed: 01-Dec-2016]

[32] Twitter, "Twitters's terms of service. Twitter terms of service," 30-Sep-2016. [Online]. Available: https://twitter.com/tos#intlTerms. [Accessed: 01-Dec-2016]

[33] Google, "Google's terms of service. Google terms of service," 30-Apr-2014. [Online]. Available: https://www.google.com/intl/en/policies/terms/regional.html. [Accessed: 01-Dec-2016]

[34] Apple, "Apple's iClound terms and conditions. V. content and your

conduct," 25-Sep-2016. [Online]. Available: https://www.apple.com/legal/internet-services/icloud/en/terms.html. [Accessed: 01-Dec-2016]

[35] "Why metadata matters," 07-Jun-2013. [Online]. Available: https://www.eff.org/deeplinks/2013/06/why-metadata-matters. [Accessed: 24-Nov-2016]

[36] J. P. Stevens, "Why you need metadata for big data success," 06-Apr-2016. [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/why-you-need-metadata-for-big-data-success. [Accessed: 24-Nov-2016]

[37] "Big data n." [Online]. Available: http://www.oed.com/view/Entry/18833#eid301162177. [Accessed: 11-Nov-2016]

[38] Wikipedia, "Big data," 11-Nov-2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Big_data&oldid=748964100. [Accessed: 11-Nov-2016]

[39] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," *Journal of Big Data*, vol. 2, no. 1, p. 21, Oct. 2015 [Online]. Available: http://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0030-3. [Accessed: 13-Nov-2016]

[40] O. R. Zaïane, *Principles of knowledge discovery in databases*. 1999, pp. 1–2 [Online]. Available: https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/. [Accessed: 13-Nov-2016]

[41] "Big data collection collides with privacy concerns, analysts say. PCWorld," 10-Feb-2013. [Online]. Available: http://www.pcworld.com/article/2027789/big-data-collection-collides-with-privacy-concerns-analysts-say.

html. [Accessed: 15-Nov-2016]

[42] "Answers.io. Answers." [Online]. Available: https://answers.io/answers. [Accessed: 14-Nov-2016]

[43] A. L. Burgelman, N. L. Burgelman, and NGDATA, "Attention, big data enthusiasts: Here's what you shouldn't ignore. WIRED." [Online]. Available: https://www.wired.com/insights/2013/02/attention-big-data-enthusiasts-heres-what-you-shou [Accessed: 15-Nov-2016]

[44] D. Laney, "3D data management: Controlling data volume, velocity, and variety," META Group, February 2001 [Online]. Available: http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume- pdf

[45] M. Hilbert, "Big data for development: A review of promises and challenges," *Development Policy Review*, vol. 34, no. 1, pp. 135–174, December 2015 [Online]. Available: http://dx.doi.org/10.1111/dpr.12142

[46] N. Davis Kho, "The state of big data," 24-Feb-2016. [Online]. Available: http://www.econtentmag.com/Articles/Editorial/Feature/The-State-of-Big-Data-108666.htm. [Accessed: 18-Nov-2016]

[47] T. C. (Apple's CEO), "A message to our customers. Customer letter," 16-Feb-2016. [Online]. Available: http://www.apple.com/customer-letter/. [Accessed: 18-Nov-2016]

[48] M. Green, "What is differential privacy? A few thoughts on cryptographic engineering," 15-Jun-2016. [Online]. Available: https://blog.cryptographyengineering.com/2016/06/15/what-is-differential-privacy/.

[Accessed: 18-Nov-2016]

[49] B. Budington, "WhatsApp rolls out end-to-end encryption to its over one billion users," 07-Apr-2016. [Online]. Available: https://www.eff.org/deeplinks/2016/04/whatsapp-rolls-out-end-end-encryption-its-1bn-users. [Accessed: 18-Nov-2016]

[50] "Stuck in traffic? Insights from googlers into our products, technology, and the google culture," 28-Feb-2007. [Online]. Available: https://googleblog.blogspot.com/2007/02/stuck-in-traffic.html. [Accessed: 18-Nov-2016]

[51] Wikipedia, "Google traffic," 25-Oct-2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Google_Traffic&oldid=746200591. [Accessed: 18-Nov-2016]

[52] "Global mobile OS market share." [Online]. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/. [Accessed: 18-Nov-2016]

[53] 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]

[54] 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]

[55] 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]

[56] M. Gaitho, "Applications of big data in 10 industry verticals," 20-Oct-2015. [Online]. Available: https://www.simplilearn.com/big-data-applications-in-industries-article. [Accessed: 19-Nov-2016]

[57] F. T. C. USA, "Personal data ecosystem," *Protecting Consumer Privacy in an Era of Rapid Change - Recommendations for Business and Policymakers - FTC Report*, March-2012. [Online]. Available: https://www.ftc.gov/sites/default/files/documents/public_events/exploring-privacy-roundtable-series/personaldataecosystem.pdf. [Accessed: 17-Nov-2016]

[58] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, p. 4, Apr. 1965 [Online]. Available: https://drive.google.com/file/d/0By83v5TWkGjvQkpBcXJKT1I1TTA/. [Accessed: 07-Dec-2016]

[59] T. Pritlove and U. Schöneberg, *Neuronale netze*. 2015 [Online]. Available: https://cre.fm/cre208-neuronale-netze. [Accessed: 06-Dec-2016]

[60] L. Columbus, "51% of enterprises intend to invest more in big data," 22-May-2016. [Online]. Available: http://www.forbes.com/sites/louiscolumbus/2016/05/22/51-of-enterprises-intend-to-invest-more-in-big-data/. [Accessed: 07-Dec-2016]

[61] "ProjectVRM - cDevelopment work. ProjectVRM," 28-Nov-2016. [Online]. Available: https://cyber.harvard.edu/projectvrm/VRM_

Development_Work. [Accessed: 09-Dec-2016]

[62] "ProjectVRM - principles. ProjectVRM," 28-Nov-2016. [Online]. Available: https://cyber.harvard.edu/projectvrm/Main_Page#VRM_Principles. [Accessed: 09-Dec-2016]

[63] The TAS3 Consortium, "TAS3 architecture - figure 2.2: Major components of organization domain." Jul. 2011 [Online]. Available: http://homes.esat.kuleuven.ac.be/~decockd/tas3/final.deliverables/pm42/TAS3_D02p1_TAS3.Architecture_final.pdf

[64] "Kantara initiative – join. innovate. trust." [Online]. Available: https://kantarainitiative.org/. [Accessed: 14-Dec-2016]

[65] T. Kirkham, S. Winfield, S. Ravet, and S. Kellomaki, "The personal data store approach to personal data security," *IEEE Security & Privacy*, vol. 11, no. 5, pp. 12–19, 2013.

[66] Y.-A. de Montjoye, S. S. Wang, A. Pentland, D. T. T. Anh, A. Datta, and others, "On the trusted use of large-scale personal data." *IEEE Data Eng. Bull.*, vol. 35, no. 4, pp. 5–8, 2012 [Online]. Available: http://sites.computer.org/debull/a12dec/a12dec-cd.pdf#page=7. [Accessed: 30-Oct-2016]

[67] "openPDS/SafeAnswers - the privacy-preserving personal data store." [Online]. Available: http://openpds.media.mit.edu/. [Accessed: 14-Dec-2016]

[68] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "openPDS: Protecting the privacy of metadata through SafeAnswers," *PLoS ONE*, vol. 9, no. 7, p. e98790, Jul. 2014 [Online]. Available: http://dx.plos.org/

10.1371/journal.pone.0098790. [Accessed: 30-Oct-2016]

[69] “Microsoft HealthVault. Overview.” [Online]. Available: https://www.healthvault.com/de/en/overview. [Accessed: 14-Dec-2016]

[70] “How it works meeco.” [Online]. Available: https://meeco.me/how-it-works.html. [Accessed: 14-Dec-2016]

[71] M. Page, “Online adver>sing – booming or broken?” Sep-2015 [Online]. Available: https://meeco.me/assets/pdf/Meeco_Case_Study_Online_Advertising-Booming_or_Broken_Sept_2015.pdf

[72] “The principles. Industrial data space e.V.” [Online]. Available: http://www.industrialdataspace.org/en/the-principles/. [Accessed: 14-Dec-2016]

[73] B. Prof. Dr.-Ing. Otto, S. Prof. Dr. Auer, J. Cirullies, J. Prof. Dr. Jürjens, N. Menz, J. Schon, and S. Dr. Wenzel, “Industrial data space - digital sovereignity over data.” Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., 17-Aug-2016 [Online]. Available: http://www.industrialdataspace.org/wp-content/uploads/2016/09/whitepaper-industrial-data-space-eng.pdf

[74] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, “Hypertext transfer protocol – HTTP/1.1,” Jun-1999. [Online]. Available: https://tools.ietf.org/html/rfc2616. [Accessed: 17-Dec-2016]

[75] M. Belshe, M. Thomson, and R. Peon, “Hypertext transfer protocol version 2 (HTTP/2),” May-2015. [Online]. Available: https://tools.ietf.org/html/rfc7540. [Accessed: 17-Dec-2016]

[76] T. Dierks and E. Rescorla, “The transport layer security (TLS) proto-

col version 1.2," Aug-2008. [Online]. Available: https://tools.ietf.org/html/rfc5246. [Accessed: 17-Dec-2016]

[77] I. Fette and A. Melnikov, "The WebSocket protocol," Dec-2011. [Online]. Available: https://tools.ietf.org/html/rfc6455. [Accessed: 17-Dec-2016]

[78] D. Crockford, "The JSON data interchange format." ECMA International, Oct-2013 [Online]. Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

[79] T. Bray, "The JavaScript object notation (JSON) data interchange format," Mar-2014. [Online]. Available: https://tools.ietf.org/html/rfc7159. [Accessed: 17-Dec-2016]

[80] J. Bradley, "The problem with OAuth for authentication." 28-Jan-2012. [Online]. Available: http://www.thread-safe.com/2012/01/problem-with-oauth-for-authentication.html. [Accessed: 17-Dec-2016]

[81] "OAuth core 1.0a." [Online]. Available: https://oauth.net/core/1.0a/. [Accessed: 18-Dec-2016]

[82] D. Hardt, "The OAuth 2.0 authorization framework," Oct-2012. [Online]. Available: https://tools.ietf.org/html/rfc6749. [Accessed: 18-Dec-2016]

[83] I. O. WG, "OAuth 2.0." [Online]. Available: https://oauth.net/2/. [Accessed: 16-Dec-2016]

[84] "OpenID connect core 1.0 incorporating errata set 1," 08-Nov-2014. [Online]. Available: https://openid.net/specs/openid-connect-core-1_0.html.

[Accessed: 17-Dec-2016]

[85] J. Bradley, N. Sakimura, and M. Jones, "JSON web token (JWT)," May-2015. [Online]. Available: https://tools.ietf.org/html/rfc7519. [Accessed: 17-Dec-2016]

[86] J. Hildebrand and M. Jones, "JSON web encryption (JWE)," May-2015. [Online]. Available: https://tools.ietf.org/html/rfc7516. [Accessed: 17-Dec-2016]

[87] J. Bradley, N. Sakimura, and M. Jones, "JSON web signature (JWS)," May-2015. [Online]. Available: https://tools.ietf.org/html/rfc7515. [Accessed: 17-Dec-2016]

[88] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976 [Online]. Available: https://ee.stanford.edu/%7Ehellman/publications/24.pdf. [Accessed: 11-Jan-2017]

[89] W. Stallings, "9.1 principles of public-key cryptosystems," in *Cryptography and network security: Principles and practice*, Seventh edition., Boston: Pearson, 2014, pp. 256–264.

[90] T. Fielding, "Representational state transfer (REST)," in *Architectural styles and the design of network-based software architectures*, University of California, Irvine, 2000, pp. 76–106 [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

[91] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, "HTTP methods," Jun-1999. [Online]. Available: https:

//tools.ietf.org/html/rfc2616#section-9. [Accessed: 18-Dec-2016]

[92] "GraphQL," Oct-2016. [Online]. Available: https://facebook.github.io/graphql/. [Accessed: 17-Dec-2016]

[93] D. Beckett and B. McBride, "RDF/XML syntax specification (revised)," 10-Feb-2004. [Online]. Available: https://www.w3.org/TR/REC-rdf-syntax/. [Accessed: 19-Dec-2016]

[94] W. O. W. Group, "OWL 2 web ontology language document overview (second edition)," 11-Dec-2012. [Online]. Available: https://www.w3.org/TR/owl2-overview/. [Accessed: 19-Dec-2016]

[95] S. Harris, A. Seaborne, and E. Prud'hommeaux, "SPARQL 1.1 query language," 21-Mar-2013. [Online]. Available: https://www.w3.org/TR/sparql11-query/. [Accessed: 19-Dec-2016]

[96] "WebID specifications." [Online]. Available: https://www.w3.org/2005/Incubator/webid/spec/. [Accessed: 19-Dec-2016]

[97] "Solid specification," 03-Mar-2016. [Online]. Available: https://github.com/solid/solid-spec. [Accessed: 17-Dec-2016]

[98] "WebAccessControl - w3c wiki." [Online]. Available: https://www.w3.org/wiki/WebAccessControl. [Accessed: 19-Dec-2016]

[99] "Databox.me." [Online]. Available: https://databox.me/. [Accessed: 19-Dec-2016]

[100] T. Heo, "Control group (v2) documentation," Oct-2015. [Online]. Available: https://www.kernel.org/doc/Documentation/cgroup-v2.txt. [Ac-

cessed: 20-Dec-2016]

[101] "Overview of linux namespaces," 12-Dec-2016. [Online]. Available: http://man7.org/linux/man-pages/man7/namespaces.7.html. [Accessed: 20-Dec-2016]

[102] "Open container initiative." [Online]. Available: https://www.opencontainers.org/. [Accessed: 20-Dec-2016]

[103] "Container runtime specification (v1.0.0-rc3)," 12-Dec-2016. [Online]. Available: https://github.com/opencontainers/runtime-spec/tree/v1.0.0-rc3. [Accessed: 20-Dec-2016]

[104] "Container image specification (v1.0.0-rc3)," 30-Nov-2016. [Online]. Available: https://github.com/opencontainers/image-spec/tree/v1.0.0-rc3. [Accessed: 20-Dec-2016]

[105] S. Spiekermann, *Ethical IT Innovation: A Value-Based System Design Approach*. CRC Press; Taylor & Francis Group, LLC, 2015 [Online]. Available: https://www.crcpress.com/Ethical-IT-Innovation-A-Value-Based-System-Design-Approa Spiekermann/p/book/9781482226355

[106] E. Dean and S. Ghemawat, "MapRednce: Simplified data processing on large clusters," 2004 [Online]. Available: https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf. [Accessed: 27-Dec-2016]

[107] T. Dierks, "The transport layer security (TLS) protocol version 1.2," Aug-2008. [Online]. Available: https://tools.ietf.org/html/rfc5246#

section-7.4.6. [Accessed: 09-Jan-2017]

[108] "Mutual authentication," 02-Sep-2016. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Mutual_authentication&oldid=737409981. [Accessed: 10-Jan-2017]

[109] "Networking 101: Transport layer security (TLS) - high performance browser networking (o'Reilly). High performance browser networking," 2013. [Online]. Available: https://hpbn.co/transport-layer-security-tls/#tls-session-resumption. [Accessed: 12-Jan-2017]

[110] P. E. Joseph Salowey H. Zhou, "Transport layer security (TLS) session resumption without server-side state," Jan-2008. [Online]. Available: https://tools.ietf.org/html/rfc5077. [Accessed: 12-Jan-2017]

[111] W. Stallings, "9.1 public-key infrastructure," in *Cryptography and network security: Principles and practice*, Seventh edition., Boston: Pearson, 2014, pp. 443–445.

[112] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, and W. Housley R. andPolk, "Internet x.509 public key infrastructure certificate and certificate revocation list (CRL) profile," May-2008. [Online]. Available: https://tools.ietf.org/html/rfc5280. [Accessed: 11-Jan-2017]

[113] "Basisleser weiterhin kritische schwachstelle des elektronischen / neuen personalausweises. Netzpolitik.org," 27-Aug-2013. [Online]. Available: https://netzpolitik.org/2013/basisleser-weiterhin-kritische-schwachstelle-des-elektronischen-neu [Accessed: 05-Jan-2017]

[114] O. Stiemerling, "Qualifizierte elektronische signatur mit dem neuen personalausweis – oder: QES mit nPA, ein selbstversuch. CR-online.de blog,"

26-Aug-2014. [Online]. Available: http://www.cr-online.de/blog/2014/08/26/qualifizierte-elektronische-signatur-mit-dem-neuen-personalausweis-oder-qes-mit-npa-ein-se
[Accessed: 05-Jan-2017]

[115] "BSI - technische richtlinien des BSI - BSI TR-03130 eID-server." [Online]. Available: https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03130/tr-03130.html. [Accessed: 06-Jan-2017]

[116] "Personalausweisportal - eID-server." [Online]. Available: https://www.personalausweisportal.de/DE/Wirtschaft/Technik/eID-Server/eID-Server_node.html;jsessionid=8C7F11821065F2505F22AFEF65F63DFB.2_cid334. [Accessed: 06-Jan-2017]

[117] K. Nguyen and C. Schwarz, "Innovatives key management für die qualifizierte elektronische signatur mit dem neuen personalausweis," *Datenschutz und Datensicherheit-DuD*, vol. 37, no. 8, pp. 502–506, 2013 [Online]. Available: https://www.bundesdruckerei.de/sites/default/files/documents/2013/08/fachartikel_dud_sign-me.pdf. [Accessed: 06-Jan-2017]

[118] D. B. der Bundesregierung für Informationstechnik, "IT-beauftragter der bundesregierung de-mail." [Online]. Available: http://www.cio.bund.de/Web/DE/Innovative-Vorhaben/De-Mail/de_mail_node.html. [Accessed: 06-Jan-2017]

[119] L. Neumann, "Stellungnahme zum elektronischen rechtsverkehr." 14-Apr-2013 [Online]. Available: https://ccc.de/system/uploads/128/original/demail_april2013.pdf

[120] N. Carlson, "Facebook connect is a huge success – by the numbers," 01-Jul-2009. [Online]. Available: http://www.businessinsider.

com/six-months-in-facebook-connect-is-a-huge-success-2009-7. [Accessed: 16-Dec-2016]