Smartwallets

Paul Trevithick, The Mee Foundation

March 3, 2023. Revised October 15, 2023

Abstract

Today's app-centric architecture for personal data has helped fuel the rapid growth of internet apps and sites. It has also resulted in a lack of autonomy, agency, and privacy for individuals. They lack the power and technical means to manage the data involved in interactions they have with hundreds of apps/sites and data privacy law has proven inadequate to prevent the disclosure of this data by these apps/sites to other actors. To address these issues we present design considerations for, and the architecture of, a personal, on-device digital wallet with an associated legal contract which together comprise what we call a *smartwallet*.

1 The status quo

We examine the status quo for individuals of life online by looking at it through the lenses of power, autonomy and agency respectively.

1.1 Power

> "The stronger becomes master of the weaker, in so far as the latter cannot assert its degree of independence –here there is no mercy, no forbearance, even less a respect for 'laws'."[15]

Let us examine how power is allocated in our society by technology and law respectively.

1.1.1 Technology

Our life online is mediated through apps and websites. In the following sections we refer to both as simply as *apps*. These apps process personal data in a few different ways: (i) data related to interactions with people using an app is stored in *accounts*, (ii) third-party adtech systems track the person and display ads on these apps, and (iii) transaction system process payment data. We discuss each of these in turn.

Account data. Apps have at their core a database. As you interact with the application, whatever you type, click, enter, upload is stored in it. This is all part of your "account." Addition observations, e.g. the kinds of things you click on, and spend time on, are also collected. In this way app's account is where your on-app interaction data lives.

Individuals have limited power over this account data. At best there may be a means to review and update it via a form. In some cases, the app allows the person to download a copy of their account data, although the process is time-consuming, cannot be automated, and results in downloading dozens of files that they probably don't know how to use. In some jurisdictions the individual has rights to rectify and erase their data, although these rights are merely theoretical. In practice, due to the impractical burden placed on the individual to exercise them, they are not actionable.

The app may sell the individuals account data to data brokers[1] who buy data from a variety of sources, collate information about individuals or groups of individuals, and then resell it.

Tracking data. Tracking is a euphemism for surveillance by businesses of individuals. It is performed through a combination of technologies implemented by the apps (e.g. third-party cookies, transparent pixels, fingerprinting, etc.) as a necessary part of their integration with adtech systems from which they derive revenue.

Tracking data is behavioral data that is used to infer traits about the individual (e.g. age-range, income level, and many of other demographic and psychographic traits). Advertisers pay to get their messages (ads, images, videos, text, etc.) in front of cohorts with shared traits (called "audiences") irrespective of which app a member of that cohort is visiting. Apps sell ad inventory (i.e. ad "slots") to these advertisers. Although some are sold directly, most are sold via ad networks and ad exchanges that take part in a high volume, high-speed real-time auction process called real-time bidding[2]. A complex ecosystem of thousands of adtech firms are involved in the supply chain stretching from advertisers, through ad exchanges, to the apps acting in the role of publishers.

Payment data. If the app sells products or services it integrates with payment gateways that allow the app to receive funds from the individual (e.g. via a payment card). In most cases this involves sending financial data (including identifiers) about the individual through financial systems that include banks and credit card associations.

In addition to the expected privacy risks associated with the regular flow of payment transactions, some apps make extra money by selling information about purchases to data brokers.

Harms In the data flows just mentioned the individual is shown to be relatively powerless

[1] theconversation.com/its-time-for-third-party-data-brokers-to-emerge-from-the-shadows-94298
[2] en.wikipedia.org/wiki/Real-time_bidding

over their data. Making matters worse, we live today in what Alicia Solow-Niederman calls an "inference economy."[23] wherein big data and machine learning are used to infer traits that form new kinds of personal data–often more sensitive than the underlying source data. Harm and risk depend upon the situation; they can rarely be determined outside of a specific situation[22]. Nevertheless, we can list representative kinds of harms:

- Individuals are vulnerable to data breaches by any of these thousands of apps.
- Individuals have no visibility into what's being gathered, where it's being shared and how it's used.
- Individuals can be spammed by marketers.
- Individuals are vulnerable to identity theft.
- Individuals can be exposed to price discrimination.
- Individuals can be exposed to from hiring discrimination.
- Individuals can be stalked.

1.1.2 Privacy

> "Debates over privacy are really debates about how power will be allocated in an information society and how much power the humans in that society will get as consumers or citizens."[18]

Today, despite significant new regulation, the basic approach to protecting privacy hasn't changed since the 1970s. It is often called *notice & consent*. Solove described it using the term *privacy-self management*, as follows:

> [T]he law provides people with a set of rights to enable them to make decisions about how to manage their data. These rights consist primarily of rights to notice, access, and consent regarding the collection, use, and disclosure of personal data. The goal of this bundle of rights is to provide people with control over their personal data, and through this control people can decide for themselves how to weigh the costs and benefits of the collection, use, or disclosure of their information.[21]

Although well-intended, and necessary, *notice & consent* does not provide people with meaningful control over their data.

The US population consistently misunderstands the meaning of the term privacy policy.[3]

[3] "Privacy policies have been widely adopted and are now commonplace. This kind of transparency is good in theory, but less so in practice since it places the onus of privacy on end users. In general, attempts to improve privacy by helping end users have not worked, since most people don't have the time, expertise, or desire to deal with all the nuances of privacy."[8]

A majority of Americans believe incorrectly the mere presence of a privacy policy indicates a website will not share information without permission.[3]

The problem is well summarized as follows:

> When presented with click-through consent, privacy policies or terms of use statements, most people reflexively select "I agree". An extensive body of academic research specifically on privacy and data collection notices demonstrates that members of the public don't read them and might not understand them if they did and that many misinterpret their purpose, assuming that the existence of a privacy policy displayed by way of notice means that the entity collecting the data offers a level of data protection when, in fact, privacy notices do not guarantee privacy. Since the terms offered are typically "take it or leave it", to decline often results in being denied the product or service one seeks, creating a disincentive for consumers to do anything other than accept the terms.[4]

"We agree to all these 'privacy notices' so we must have privacy, right? Notice and choice is thus an elaborate trap, and we're all caught in it."[18]

Progress: GDPR and CPRA

The most substantive lever for progress has been legislation such as CDPR and CalOPPA, along with regulatory fines by organizations like the FTC.

In a growing number of jurisdictions, including Europe under GDPR[4] and California under CPRA[5], the person's *data rights*, (e.g. the right to access, recitfy and erase their data), are clearly described. Unfortunately in practice the time and effort required to exercise these rights at each app individually is enormous. The individual must, for example, send written requests to get copies of their data, update it, or have it be deleted. Until these processes are automated by personal agents, in practice these rights don't exist.

Privacy and protection of children

Society agrees to supervise the places children inhabit, protect them from environments they should not encounter, and regulate the products they use. As a result, businesses are not permitted to sell tobacco, alcohol, pornography, handguns, certain kinds of fireworks, and other products and services to minors. However, none of this is true online. In the virtual world children are largely unprotected despite being exposed to wide range of potential harms.

Many approaches have been proposed and tried without much success. Existing laws have proven to be insufficient, and industry self-regulation has largely failed. Today there is a

[4] gdpr-info.eu/
[5] thecpra.org/

renewed global push to protect children's safety through stronger laws and regulations. Although some use other approaches[6], many mandate age verification.[6][9] However, privacy advocates and others have shown that many of the mechanisms for verifying age online weaken anonymity and privacy.[19]

1.2 Autonomy

Definition: *freedom from external control or influence; independence.*[7]

1.2.1 Independence

We each have a self that embodies our unique individuality. We "bring" that independent selfness to our interactions with others. However, online "we have no *digital embodiment*."[8] Our identifiers and their associated account data are provided to us by online service providers (e.g. in the form of a Facebook or an Amazon account) and without them, we don't exist. We can't "bring" them anywhere. Anyone who has been banned from a platform, or uses a platform that has been shut down, is sharply reminded that their account and its data exists at the pleasure of that platform.

Note: Since our discussion applies equally to an online service provider's mobile app, webapp, or website, we will simply use the term *app* to refer to all of them.

1.2.2 Ownership

We all share a simple, intuitive sense that our personal data is *ours*[9]. We believe this even if it is almost always in the possession of external organizations as a by-product of our interactions with their apps. Despite not being held by us, we expect that it to be treated with care. This expectation is so fundamental that it feels like a human right.

This sense of ownership assumes we have certain rights over our personal information wherever it may reside. Indeed, privacy laws such as GDPR[10] provide the individual the following rights over their personal information:

- Right to personal data transparency
- Right to rectify
- Right to restriction of processing

[6]Such as requiring online services that are likely to be used by young people to default to the highest privacy setting possible for minors, as mandated by California's Age-Appropriate Design Code Act.

[7]languages.oup.com/google-dictionary-en/

[8]Phil Windley, personal communication, September 2022

[9]Let us for the moment put aside the complex legal and technical issues related to the ownership of data

[10]gdpr-info.eu/

- Right to access
- Right to object
- Right not to be subject to automated decision making, including profiling
- Right to erasure ('right to be forgotten')
- Right to data portability

These laws are largely impotent in practice because without APIs on the provider's side and personal tools to consume them on person's side, the burden required to exercise these rights is too heavy for anyone to bear.

1.2.3 Possession

In theory, and as we have just discussed, ownership doesn't require possession. That is, with sufficiently strong legal mechanisms (some of which we will propose later in this paper) a sense of ownership can be provided irrespective of where our data is stored and by whom.

In practice possession tends to shift power to the possessor. Unfortunately, with few exceptions our personal information is stored and managed by service providers. This pattern of what could be called *app-held data* by the *first-parties* we interact with is so common that it's hard to imagine an alternative. Beyond first-parties, our data is also collected and held by *(*third-parties) (e.g. data brokers) with whom we have no direct interaction. In short, as Johannes Ernst has put it, "everybody has our data ... except us."[11]. Giving people possession of their data doesn't mean that it doesn't also exist in many other places, but what it does mean is that *at least* we too have it!

1.2.4 Lock-in

Our account identifiers and associated human data are bound to specific online service providers and can't be moved freely from one to another. They are not portable.

In many jurisdictions online service providers are required by law to provide us with access to our data, but they usually offer this access by means of a set of files emailed to the person as an attachment hours, or days after the request. There are serious problems with this kind of access. First, it is tedious, manual, and slow. Service providers don't support data "export" APIs, so an individual can't use technology to automate the process. Second, the individual ends up with dozens of sets of files (one set from each provider) that are not human-friendly and largely unintelligible.

[11] reb00ted.org/personaldata/20210620-who-has-my-personal-data/

Beyond access and export problems, providers generally don't provide "import" APIs to allow the person to upload their data to another provider. Thus, even if a person could upload their data to another provider, it would have to first be transformed into the format of the recipient, since each provider uses their own format. In the end we lack data portability for our own data.

Advocacy groups including the EFF are pushing for interoperability on the internet as an antidote to corporate concentration. This is good as far as it goes but they need to go farther. They should insist that apps implement import/export APIs that can be leveraged by agents such as smartwallets. "A new regime of interoperability can revitalize competition in the space, encourage innovation, and give users more agency over their data..."[1]

1.2.5 Peer-to-peer

We lack the ability to directly communicate person-to-person (e.g. chat) with others without having to rely all parties having accounts on the same server. With a few exceptions[12], we don't have the ability to do so *peer-to-peer*–i.e. from on person's device to the other person's device. Instead, we're dependent on servers hosted by intermediaries. Further, whereas it is now standard practice that the content of messages is end-to-end encrypted, the metadata about them (e.g. who a person communicates with, from where, at what time, how often and from which device, etc.) is in many cases visible to the intermediary server.

1.3 Agency

Definition: *the capacity, condition, or state of acting or of exerting power*[13]

2 Related work

Many initiatives have arisen to address the challenges we've described. We mention a few of them here.

The lock-in and lack of data portability and interoperability between service providers is being fought using both policy and technical means[2][1].

Work related to re-decentralizing the internet include: rececentralize.org[14], DWeb prin-

[12] berty.tech
[13] www.merriam-webster.com/dictionary/agency
[14] redecentralize.org

ciples[15], The Web3 Foundation[16], the Decentralized Identity Foundation(DIF)[17], "local-first" software principles[18], ProjectVRM[19], Blue Sky[20], and Berners-Lee's Decentralized Information Group[21].

Relevant is work on *personal agents* [22]–software tools that work (i.e. provide agency and power) "on the individual's side"[24] for, and *exclusively* on behalf of, the person. Personal datastores and the *self-sovereign identity*[17] movement are squarely aimed at addressing our lack of autonomy.

Personal data ownership include: "user-held" data[10], where your data is held by you in a personal datastore[25].

Also relevant (and inspiring) is work on "local-first" software.[12]

3 Design considerations

In this section, we discuss design considerations for solutions that intend to address the symptoms described in the previous section.

3.1 Human-centricity

Many of the challenges described thus far have their origin in an architecture that is *provider-centric* rather than *human-centric*. The internet includes millions of providers, each offering their own app[s]. In this provider-centric model each provider's app sees a narrow slice of the individual through the lens of their direct interactions with them.

For the individual the situation is reversed. They sit at the center of a hub with many dozens of connections to apps radiating outwards from them. Even for a single app there is considerable burden for the person to enter and update personal information, payment details, and preferences, and review privacy policies, and set cookie preferences, and so on. Multiplied by perhaps one hundred connections the resulting burden is practically impossible.

[15] getdweb.net/principles/
[16] web3.foundation/
[17] identity.foundation
[18] inkandswitch.com/local-first/
[19] projectvrm.org/
[20] blueskyweb.xyz/
[21] dig.csail.mit.edu
[22] What Mozilla calls a *user agent*[23]
[24] Project VRM[20] refers to this as "tools for individuals to manage relationships with organizations" to which we would add "...or with other individuals."
[25] Examples of open-source personal datastores include `https://solidproject.org`, Decentralized Web Nodes(DWN). For more about personal datastores see `https://wikipedia.org/wiki/Personal_data_service`

Tools to manage these chores must sit on the person's side, and work on their behalf across all of them. Technologies of this kind, that empower a person across multiple apps, e.g. browsers and password managers, are called are *user-agents* since they act as agents of the person.

3.2 On-device storage and processing

If we assume a human-centric decentralized architecture, where should the person's personal datastore and associated processing live? Should it be on-device or in the cloud? By *on-device* we mean that the primary location for a person's datastore and processing is on their own phones, laptops, and perhaps home servers. Cloud-based means that the person's datastore and processing lives primarily in the cloud (e.g. on a SOLID[26] pod). We say *primarily* because there are usually use-cases that involved replicating/syncing some of the data to the "other" location. The local-first software[27] principle are highly relevant.

Security. Although this is debatable, it is our contention that given a large number of people, having a personal datastore on-device is more secure than in the cloud. Even if each alternative were equivalently secure for a single person, a cloud-based architecture by its very nature aggregates large numbers of personal datastores at one cloud service provider location and thereby creates a much larger economic incentive for hackers.

Equity. Any solution must be able to be afforded by all socio-economic classes and not just those better off. For this reason, we believe solutions that incur monthly hosting fees are disqualified. Since people own their devices and can "host" new apps there, the situation is better, although there is a cost for the additional storage required for an on-device datastore.

3.3 Replication

If we assume most of a person's information is held on-device, we need to solve the roaming problem when a person has two or more agents running on multiple devices each of which is only intermittently connected to the internet. The person's data needs to be kept consistent across these agents and devices, at least eventually. This requires that the person's agents implement data replication and syncing between themselves in a peer-to-peer (P2P) fashion. Unfortunately pure P2P communication between agents running on differing device platforms remains an unsolved problem and intermediate relay servers are required.

Since relays are a necessary part of the deployment architecture, for privacy, autonomy, trust, and security reasons they are subject to their own design considerations. We touch on a few of them here. For the very few people who are able and willing to self-host their

[26] solidproject.org

[27] www.inkandswitch.com/local-first/

own relays, the relay needs to be free, open-source and easy to build and deploy. Everyone else will have to trust some external administrative authority. Hopefully relays will be available freely or at very low cost. In either the self-hosted or external case, the relay needs to be trusted. For this reason its source code should be open and the relay should only store encrypted data. It should store message data only while waiting for the recipient wallet to come online.

3.4 Backup

One disadvantage of *non-custodial*, on-device architectures (as compared to cloud-based architectures) is the vulnerability wallet owners who are not diligent about backing up their devices (e.g. to an online service) face of losing their wallet-managed personal data. For people with more than one device this is less likely since data is replicated (as mentioned above) across their devices, and a repaired or replaced device's data can be restored from one of the person's other devices. There remains of course the worst-case scenario wherein the person hasn't backed up any of their devices and all of them are lost or damaged simultaneously.

Agents could implement backup/restore approaches that would provide recovery from even this disaster, however they are themselves complex and problematic. The wallet's data must stored in remote storage location(s) and encrypted using a master passphrase that the person must never be able to forget or lose. To do this, various approaches including sharding, shared secrets[28], and social recovery have been proposed but this is still an emerging area.

3.5 Loyalty

Much of the power asymmetry described in the first section is due to economic incentives for online service providers. These incentives motivate them do just enough in the person's interest to keep them using the service and generating personal data that the provider can monetize. Personal data, after all, is now considered to be an asset class–the more of it that's collected the better. For a smartwallet to work *exclusively* on behalf of the person, the *smartwallet provider* (i.e. organization that develops and publishes it) must not have an economic incentive to be disloyal to the person's interests. One way to ensure this is to design the wallet such that its data remains within itself and is never stored by, or even accessible by, the wallet provider.

3.6 Metacontextuality

Zuckerberg once said that "[h]aving two identities for yourself is an example of a lack of integrity"[11]. However, even if one could force everyone using a single platform (e.g.

[28] en.wikipedia.org/wiki/Shamir%27s_secret_sharing

Facebook) to have a single identity[29], this approach is clearly unworkable for a solution that represents the person across multiple, widely varying systems and contexts. People need the freedom to be themselves–selves that are complicated and messy. Our identities vary depending on whom we are interacting. We choose to express different parts of ourselves within different contexts. Not only are the attributes we share different the values of one attribute may be different in different contexts.

> "[A]t various times in the same day, virtually every adult can be a friend, a worker, a supervisor, a citizen, a mentor, a student, a musician, a customer, a lover, a child and a parent. Each of these roles demands different behavior and different aspects of our selves, aspects that need not be consistent. We behave, for example, in different ways with loved ones than with those we encounter in commercial or professional settings. Even among our loved ones, we behave very differently (and often show very different sides of ourselves) to our children, our parents, and our sexual partners. But this is not dishonest, nor is it inconsistent. At the very least, it's no more inconsistent than is the complicated nature of having a self. It is human."[18, p122]

Let's look at a person's age as an example. We see that across contexts they might share, their exact chronological age among their close friends, a fictional age to a music recommendation service, no age at all in contexts wherein doing so might cause discrimination against them, or a merely a statement that they exceed the legal drinking age.

In his last public speech[30] Kim Cameron[31] introduced two useful definitions based on archaic English:

- **Selfness**: The sameness of a person or thing at all times or in all circumstances. The condition of being a single individual. The fact that a person or thing is itself and not something else. Individuality, personality.
- **Whoness**: Who or what a person or thing is. A distinct impression of a single person or thing presented to or perceived by others. A set of characteristics or a description that distinguishes a person or thing from others.

Figure 1 illustrates these concepts and introduces the notion of context.

Using these terms we can say that in everyday life people have one *selfness*, but they have many, context-dependent *whonesses*. Any solution must be meta-contextual–it must embrace and support the complicated, multi-contextual nature of our lives.

[29] Note: *identity* is term we prefer to avoid due to its semantic ambiguity, but this is the word he used.
[30] www.youtube.com/watch?v=9DExNTY3QAk
[31] en.wikipedia.org/wiki/Kim_Cameron_(computer_scientist)

Figure 1: Multiple whoness-contexts around a single selfness

3.7 Delegation

In *A Human Rights Approach to Personal Information Technology*[7] Gropper asserts that there is an architectural principle that must be adhered to in order to respect human rights [e.g. to privacy]. He identifies three universal components:

- **Authentication** (signing-in and signing documents)
- **Request** for information (e.g. forms, searches, conversations)
- **Storage** (e.g. labs, prescriptions, social contracts, transactions [, other human information])

The then asserts what could be called the *Gropper Principle* as follows (our words, his ideas):

> "Any system that respects the human right to privacy must not bundle authentication, request, and storage."

In his presentation[32] at the 2022 Identiverse conference provides additional detail (see slides[33]). It explains that only a decentralized architecture can implement the Gropper

[32] identiverse.com/idv2022/session/841489/
[33] drive.google.com/file/d/1lwaMVkG4kLi7z6cXhqMx-DGkUww9azW3/view

Principle because each of the three components needs to be implemented separately. For this to work in an open world with multiple alternative component providers, there will need to be a convergence on open standards between these three components.

3.8 Trustworthiness

Any smartwallet manages highly sensitive information. In order to be adopted voluntarily by people any solution must be trustworthy–people must have confidence that their information isn't being used against their interests.

The transparency of open-source software can to help build confidence that the technology is trustworthy. In open-source software the source code is visible to anyone to review and audit to ensure that the solution is secure, free from vulnerabilities, and works in the person's interest.

In addition to open-source, people will also consider the nature of the organization offering the solution as to trustworthiness. The organization's financial incentives should be aligned with their member's interests. A nonprofit organization could be formed which has no financial or business incentive to exploit their member's data against that member's interest. Ideally the organization would not need to have any access to personal data and thus no need to have to trust them, their security infrastructure, their processes, etc.

3.9 Data Governance

Once data is shared from the smartwallet to a first-party there are no technical means to constrain what the recipient can do with it. No technical means, for example, can prevent them from selling it others. Instead, legal means must be employed. Existing privacy regulation is insufficient, so we propose that first-parties sign a Human Information License (HIL) to license the person's information. The HIL terms are fair and balanced, and respect the person's privacy rights. This contract is signed by a trusted organization[34] that represents the community of smartwallet owners thereby making the processes effortless for them. Lastly, this organization is responsible for enforcement of the contract's terms, again, on behalf of the individual.

3.10 Data Rights

In many jurisdictions people have *data rights* to access, correct and delete their own personal information managed by providers. Privacy regulations state these rights, but in practice the burden of exercising them across hundreds of apps is unmanageable. People need agents that can automate these processes, and thereby reduce the amount of work to a practical level.

[34]These kinds of organizations have been variously described in the literature as "data unions," "data coalitions," "Mediators of Individual Data" (MIDs) by Lanier et al.[13], etc.

4 Smartwallets

We now propose a solution to the problems described in the first section that takes into account the design considerations in the second section. What we call a *smartwallet*, gives individuals control over their personal information as they interact with websites, mobile apps, and other people's smartwallets through a combination of technical and legal mechanisms. It combines a legal contract and a trusted, personal agent[35] with a traditional digital wallet[5].

A smartwallet is an app that runs on a person's devices (e.g., mobile phone, laptop, etc.) where, entirely under their control, it maintains a local, private database of their personal information. By default, none of this information is shared with any other entity, including the organization that develops the smartwallet and provides ancillary services to support it. When an app wants to know something about the person, the smartwallet shares as much or as little as the person chooses.

As with traditional digital wallets, apps/sites can request personal information from and provide information to a smartwallet. But app/sites can, if they wish, go further, and chose to agree to the terms of a Human Information License (HIL). This agreement requires that they abide by certain privacy principles in how they handle the person's data (e.g. requiring explicit consent for collection, processing, storage and sharing of the person's data) as well as implement a new API. This API enables what we call *private sharing* between the smartwallet and the app/site. Private sharing allows the person to share personal information with confidence that it remains under their control. The heart of the HIL is the concept that the person licenses their information to the app/site rather than transfering a copy of it and blindly hoping that the app/site will treat it with care. The person can exercise their rights to access, correct and delete their information stored on the app/site by using a smartwallet that connects to this API. App providers that agree to the Human Information License can become *certified* by the smartwallet provider.

4.1 Benefits for the individual

4.1.1 Privacy

When an individual has a smartwallet, they become the authoritative source of information about them. Apps, acting as a "first-party" can request information from the wallet under the terms of the HIL contract. By default, the app can't sell, transfer or share the individual's information without their consent. If a HIL contract is not in place, the wallet by default sends Do Not Sell signal to the app.

As wallet usage increases the need for data brokers is eliminated along with all the privacy threats they entail. Similarly, smartwallets's ability to generate profiles on-device

[35]Similar ideas have been proposed by others. See *personal user agents*, in [4, p24]

eliminates the need for internet tracking and the most harmful aspects of the surveillance advertising business model.

Lastly, we envision that minors can be given smartwallets by their guardians that can provide them with an age-appropriate experience online. The guardian would register their minors on a third-party age verification service and issue into the minor's wallet an age verification credential. When the minor shows up at an app it can signal that it would like to have an age appropriate experience. In response the app can request the age verification credential from the minor.

4.1.2 Autonomy

A smartwallet provides the individual with a digital embodiment independent of any app. Over time, it enables them to build rich context-specific profiles about themselves. This gives the individual a sense of ownership over their digital embodiment.

Smartwallets reduce lock-in, because they provide data portability. They provide a convenient way for the individual to retrieve their information from one app and share it with another.

As the community of wallet holders grows, surveillance free, end-to-end encrypted communications could be implemented between them. Ideally these communications would have minimal reliance on cloud-based relay servers which may be needed to buffer messages to endpoints that are temporarily offline.

4.1.3 Agency

A foundation for Personal AI

Rather than have to trust a shared AI-in-the-cloud service with all of our sensitive personal information, a better architecture has *Personal AI* algorithms run on the individual's devices. These agent-algorithms read and write personal information to/from the person's smartwallet.[36]

Logging in without passwords

Smartwallets enable the holder to log in to apps using a variety of password-less authentication technologies. Since the wallet knows who the holder is (because the holder authenticates to the wallet), the wallet can represent that person in their interactions with apps and do so without revealing correlatable identifiers. This is both more secure and more convenient.

Wielding credentials

[36] Iron Man's J.A.R.V.I.S.[37] is an example of this architecture and offered Iron Man complete privacy.

In real life you can present your driver's license to a wine seller in order to prove that you are of drinking age because the wine seller trusts the license issuer. The interaction is privacy-respecting because the presentation interaction is not disclosed to the issuer. Your driver's license could be described as *wielding* a trust credential. Unfortunately, there is no equivalent way to do this online, at present. There's no standard way to be issued a credential, hold it in a digital wallet, and then present it to another party. With a few, domain-specific exceptions (e.g. cryptocurrency), there is no standard online method for you to prove something one party states about you, to another party. Luckily digital wallets are rapidly emerging to meet this need and this capability will be included within smartwallets.

Automated data presentation

Apps rely on form filling and other kinds of (tedious) manual data entry because individuals lack the ability to *digitally* present personal information–we're carbon-based life forms, not digital! Instead, we have to manually re-enter it each time on different apps, endlessly repeating ourselves. We lack an agent that can present information on our behalf.[38]

With a smartwallet the person *never has to repeat themselves* as they move from app to app across the internet.

When using apps, people are often asked to provide information about themselves that another app has already asked them, such as "what is your email address?" This is a symptom of the internet's silo-ed architecture wherein each app maintains its own database of personal information. The person has the hassle of repeated data entry, and the app offers a less-than-optimal user experience.

Our inability to present ourselves digitally is a contributing factor to the corporate concentration on the internet. For example, it's simply easier to buy something from Amazon because so many of us have already entered so much information to them. We have a preferential attachment to Amazon that goes beyond their intrinsic advantages. Smartwallets that can represent an individual to any e-commerce website (sticking with the shopping example) and provide the same Amazon-like, frictionless UX will chip away at these "natural" monopolies.

Infer and present ad profiles

A smartwallet can generate on-device an ad profile by inferring traits from your browsing behavior. The wallet holder can review and edit this profile and may choose to share it with apps that are supported by interest-based advertising. This approach eliminates the need for surveillance by third-parties using cookies and other tracking technologies. It is similar in design to Google's Topics API[39].

[38] The credential presentation interaction just mentioned is another example of this.

[39] developer.chrome.com/en/docs/privacy-sandbox/topics/overview/

Delegation

In the offline world one entity can grant access to some resource to another entity. For example, an individual can give their car keys to a friend, so they can borrow their car. There is no standard, secure way to do this online. This is especially problematic in healthcare scenarios where a healthcare provider needs access to electronic health-related data about a patient, whereas the patient may not be able to provide it by themselves but instead needs to rely on someone else, e.g. a family member to grant the needed permission. In the online world each service provider not only possesses your data, but they do so in such a way that it is impossible for you to delegate rights to it to others.

Content filtering

Social networking platforms have replaced human content editors with algorithmic filters. People might think that they see a balance of content whereas in reality they are trapped in what Pariser called "filter bubbles."[16] Pariser's recommendation is that if platforms are going to be gatekeepers, they need to program a sense of civic responsibility into their algorithms, they need to be transparent about the rules that determine what gets through the filter, and "they need to give user control of their bubble."[14, p66]

Password management

The average person uses roughly 100 websites and 25 apps daily. Although managing and periodically updating strong, unique passwords at each is impractical without an automated password manager, it has been estimated that less than five percent people online use one.

Account management

The person shoulders the burden of maintaining the timeliness and consistency of their account information at hundreds of apps. For example, updating contact or credit card information at each is tedious, time-consuming and encourages the person to spend more time at sites that already have their information. The relative convenience of shopping on Amazon vs. other e-commerce sites is a consequence partially caused by the person not having a smartwallet to manage these relationships.

5 Design notes

Although the smartwallet is an interactive application, it operates in the background most of the time. Working solely in the person's interest, it collects information from apps that already hold their data and shares it with other apps that need it.

A smartwallet is multi-protocol. We illustrate with a few examples. If an app wanted to know the person's email address, it might ask for it in a web form. In this case the

smartwallet would use its form-filler "protocol" to fill in the value. If the app supports password-less sign-in (e.g. using OpenID Connect) the smartwallet acts as the so-called *identity provider*. If an app needed a digital driver's license credential, the smartwallet acts as a digital wallet and presents this credential that it had presumably downloaded earlier from an issuing app. In these different examples, different protocols for information sharing would be used, and the smartwallet must be technology-agnostic and support all of them. Only in this way a smartwallet be human-centric and put the person at the center of all of their online relationships.

5.1 Self and Contexts

The smartwallet represents both the person's single *selfness* and their multiple context-dependent *whonesses* that they have in their interactions with other apps and people.

The selfness of the person is held in a data container called the *self*. The contents of the self are holistic and therefore quite sensitive. For this reason, they would normally not be shared in a direct or comprehensive form with others. The person's self is the point of integration across contexts each of which may use differing identifier namespaces, protocols for communication, and schemas for knowledge representation.

Each context is represented by a *context* data container. A directed *correlation* link points from an entity in the self to the entities representing the person in each context. To ensure privacy only the person knows that each of these separate contexts contain representations of them. Each context represents an interaction via some communications protocol with an external app, website or smartwallet.

We can illustrate these concepts with a simple example. A person might play a game on a gaming app using the id DevilSpawn666, while communicating on Twitter as @alicewalker and subscribing to the Olde York Times as alice.walker@gmail.com. Figure 2 shows a simplified view of how this is represented:

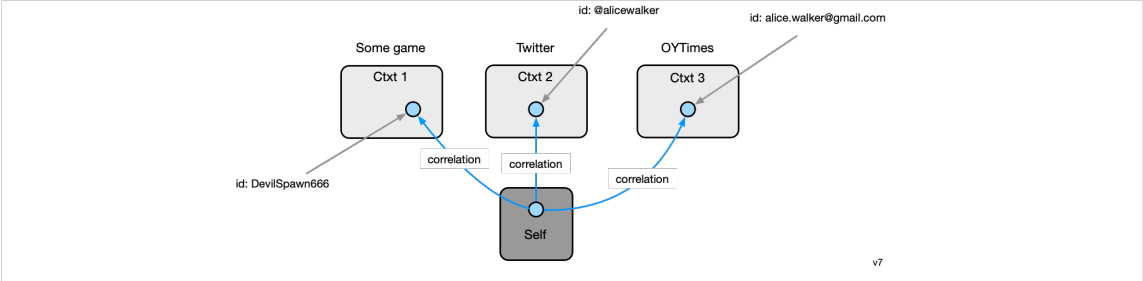

Figure 2: Alice with three contexts

5.2 Functionality

Figure 3 shows a summary of the functionality of a smartwallet.

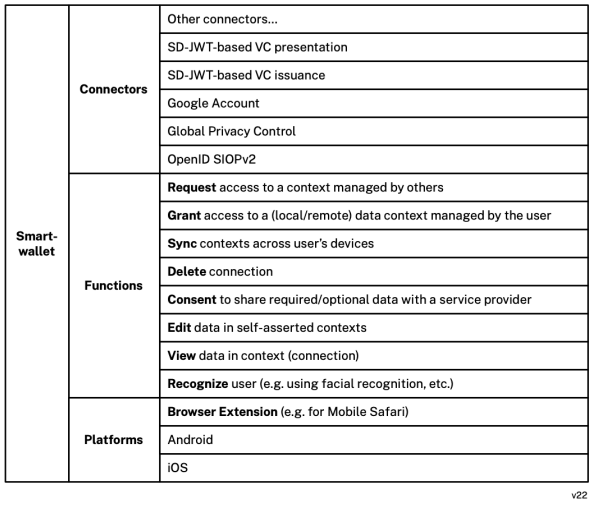

Figure 3: Smartwallet functionality

5.2.1 Connectors

A smartwallet is a tool to allow its user to manage data *connections* with apps and agents of other people. Due to the different communication protocols and data storage approaches involved in these connections, agents use an extensible architecture that leverages a set of *connectors*.

Here are a few examples of connectors:

- SD JWT-based VC presentation - present Verifiable Credentials from the smartwallet
- SD JWT-based VC issuance - store a Verifiable Credential in the smartwallet
- Gooble Account - pull data from myaccount.google.com
- Global Privacy Control - sends a "Do Not Sell My Personal Information" signal to apps
- OpenID SIOPv2 - allows the person to authenticate with an app without using passwords, without first creating an account, and with surveillance by an external so-called *identity provider*.

5.2.2 Functionality

Here are functions exposed to the person through the UI:

- **Request** access to another person's information
- **Grant** access to selected portions of your information to another person
- **Sync** contexts across person's devices
- **Delete connection** delete all data associated with this set of contexts
- **Consent** to share required/optional data with a service provider
- **Edit** data in self-asserted contexts within a connection
- **View** data related to a connection
- **Recognize** the smartwallet owner (e.g. using facial recognition, etc.) and thereby prevent others from using the smartwallet

5.3 Architecture

In this section we present the architecture of a smartwallet. The multi-layered architecture of Alice's smartwallet is shown in the center of Figure 4. We concentrate here on the smartwallet itself and leave a discussion of its interactions with the four apps (one to the left and three to the right of the smartwallet) to section 6.2 later on.

This section references terms such as *self*, *context*, *connection*, and *protocol* that are described in section 5.4 where we describe the smartwallet's data model.

5.3.1 Platform SDK

A wallet is deployed on one or more of the person's devices (e.g. a smartphone, laptop, etc.) and is developed as a native application (e.g. written in Swift on iOS, Kotlin on Android, etc.) using the platform's SDK. The User Interface (UI) component provides a user interface to manage a person's data sharing relationships with apps. Using this UI the person can add and delete connections. Within each connection they can consent to data shared from their wallet, see what data is involved in the connection, and edit attribute values–at least in in non-VC related use cases where self-asserted data is acceptable.

5.3.2 Controller layer

The controller layer includes a Foreign Function Interface (FFI) component that handles requests from the UI. It translates these requests into the Controller component. The Controller updates context attributes via the Data Access component based on commands

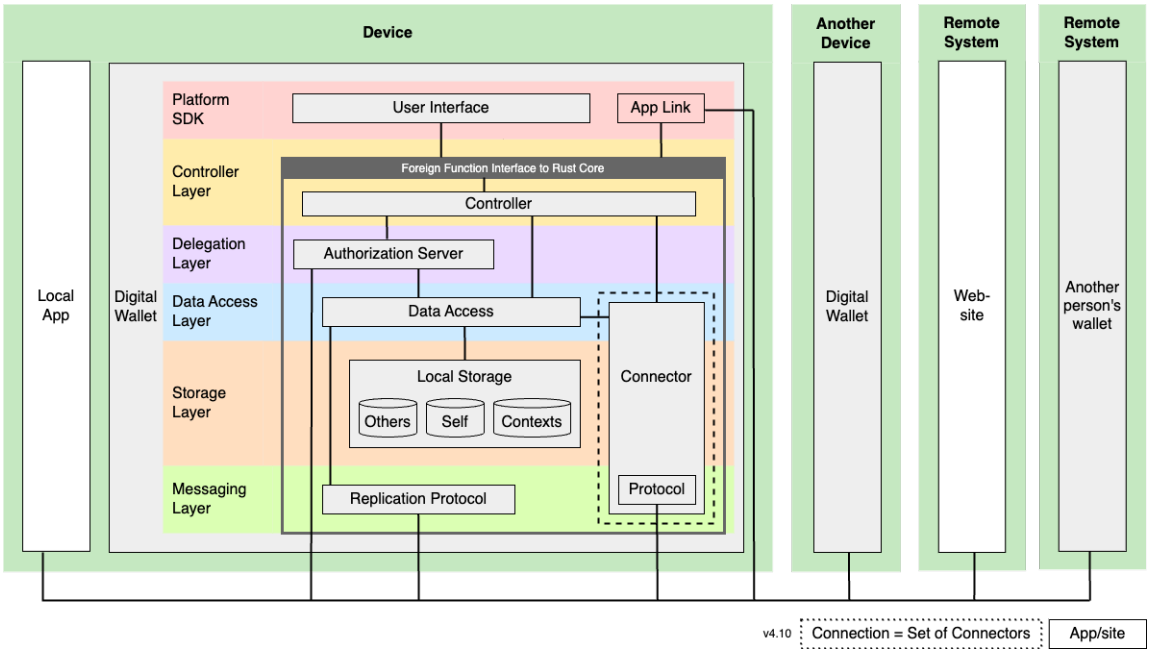

Figure 4: Smartwallet Architecture

from the FFI. Lastly, the Controller is responsible for creating and deleting connections and their component *connector(s)*.

5.3.3 Delegation Layer

The *authorization server*[40] (AS) responds to requests for access to the person's contexts–contexts which are managed by the data access layer described below. The AS sends these requests to the controller layer (which may in turn call back to the UI Layer) to allow the person, either interactively or by policy, to grant or deny them. These requests can come from other people's smartwallets and other entities that support the smartwallet's delegation protocol.

5.3.4 Data Access Layer

The Data Access component is responsible for management of the holder's data. It exposes it via the Controller to the User Interface where it can be viewed and in some cases edited. The Data Access component provides replicated data storage for Connectors. It relies on the Local Storage component and sends change sets to the Replication Protocol component in the Messaging Layer which broadcasts them to the person's other wallets.

[40] For example, a GNAP `oauth.net/gnap/` Authorization Server

5.3.5 Storage Layer

This layer uses the Local Storage component to persist (i) the state of Connectors (e.g. storing VCs in connections with issuers or verifier relying parties (in *Contexts*)) (ii) other wallet holder state (in *Self*) and (iii) metadata about relying parties (in "Others"). Data in the Local Storage component is encrypted using FIPS-compatible algorithms.

5.3.6 Messaging Layer

The messaging layer consists of a set of libraries for various communications protocols. Two kinds of protocols are shown in Figure 4. The first kind is the built-in Replication Protocol used to replicate the person's wallet state across their devices. The second kind are protocols implemented by connectors to communicate with relying parties. All data is encrypted in transit using FIPS-compatible algorithms.

5.3.7 Connectors

Now that we've described the horizontal layers of a smartwallet, we turn to the *connector* extension point. A smartwallet typically has multiple *connections* each of which is implemented by one or more *connectors*. Each connector has a communications aspect and a storage aspect.

Communications aspect. A connector implements the communications protocol used by the other party (e.g. a service provider or another person's smartwallet). We use the term protocol very loosely since the nature of the other party varies considerably. It could for example be an API of a service provider, another person's smartwallet, an authentication protocol exposed by an endpoint, or something entirely different.

Storage aspect. A connector persists its state in a context container managed by the Data Access component. The Data Access component requires this context state to be represented in the Persona data model, so the connector is responsible for dynamic, bi-directional schema transformation between the data model of the protocol and the Persona data model.

5.4 Persona Data model

This section describes the data model of a smartwallet. The person's data adhering to this model is replicated across two or more agents running on different devices, but we focus here on the logical model, not its replicas. The data model can be thought of as a three level hierarchy of data containers each of which holds *Person* instances representing the user. The top layer consists of a single *Self* container. The middle layer are *Group* containers. The bottom layer consists of *Context* containers.

These Person instances are connected into a directed graph that spans these three levels of containers. The singleton Self container holds a single Person node that represents the selfness of person as a single individual. The Self has a set of context containers each of which represents how the person is presented to or perceived by another party (e.g. another person's smartwallet or a digital service provider's app)–that is their whoness. Note that any number of combinations of communications protocols, local apps and web services may be involved in the connection between the smartwallet and another party. The Person node in the self container has no scalar attributes but usually contains a set of correlation links pointing to a corresponding Person node in multiple contexts.

Between the Self and the leaf context containers may be a set of intermediate level Group containers. These Groups also contain a Person node representing the smartwallet owner. This Person node is linked to"sub" Person nodes in the child containers of the Group container. It may also have attributes of its own. The Person node in a Group container can be used to represent a role the person might play in a set of child contexts.

In the simplified example shown in Figure 2 a person, Alice, whose selfness is represented by a blue person node in the Self context. Alice has a relationship with three other parties: a game, Twitter, and the Olde York Times. Each of these relationships is represented by a context. The whoness, or facet of Alice that she exposes in each context is represented by a person node in each of these three contexts.

The information in a context (most importantly Person nodes) is read and written to by the smartwallet based on the data flowing through the smartwallet's connection with the other party (or more precisely, with the apps of the other party). We have added three of these other parties explicitly to Figure 5, and added a new kind of container, called Others, to contain them.

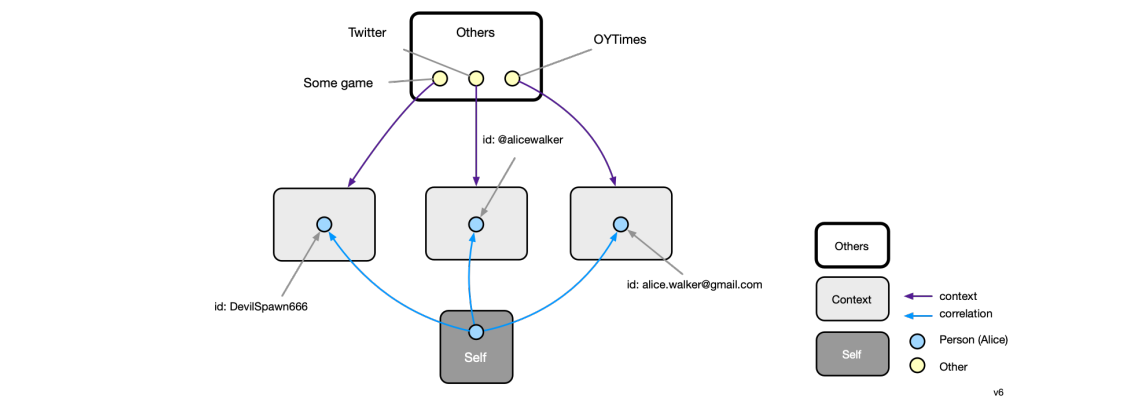

Figure 5: Alice's Self and Others

The personal data flowing through each of the three connections represented by the purple lines above may flow from the smartwallet, to the smartwallet, or in both directions. It may have originated on either side. It may be self-asserted claims (attributes) entered by the person directly into the smartwallet. Or it may be claims entered by the person on an app of the other party, or sensed by a local app (or sensor), or generated by the other party based on direct on-site or on-app interactions with the person.

5.4.1 Container classes

We describe the data model in two parts. The first part describes the data containers. The second describes the data held by these containers. Let us start describing the data model of the containers themselves. Figure 6 shows the various data container classes.

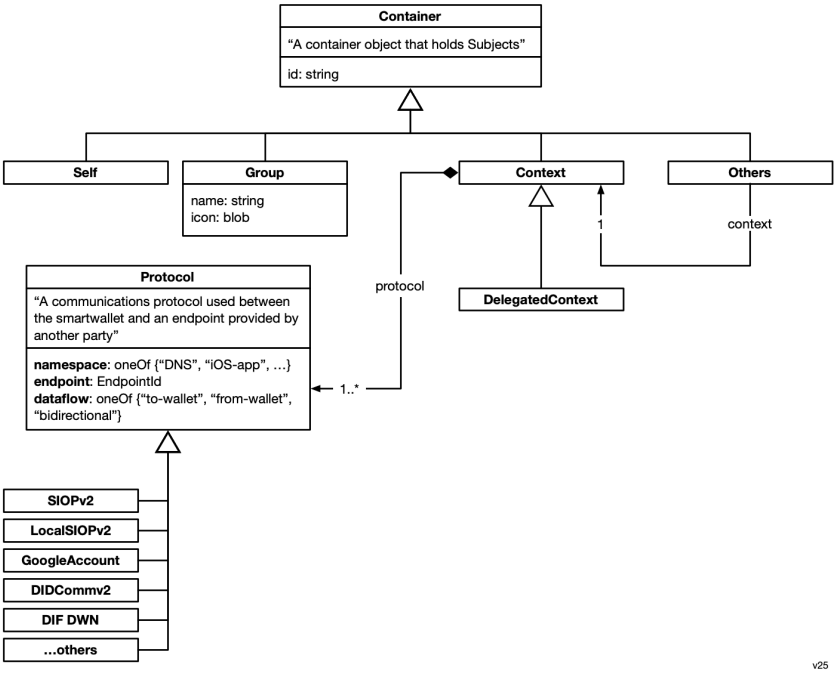

Figure 6: Container classes

Classes

- **Others** - a container holding a set of Other nodes (see Persona Classes). Each Other node represents a party with the person has a connection. These Others may be other people or legal entities. If they are legal entities, they are often called relying parties, such as a digital service provider like Twitter, Inc. Each other has the following properties:

  - **Context** - a single Context that captures one aspect of the overall connection

- **Self** - the single Container holding a single Person node that represents the selfness of the person
- **Group** - an intermediate level container that holds a single Person node that represents a common role or persona that the person plays. A Group has these attributes:
  - **name** - the name of the group
  - **icon** - an icon for the group
- **Context** - a Container holding a Person node that represents the person in a specific aspect of their relationship with some other party. We say "specific aspect" because the relationship between the person a given other, may be represented by more than one context, each representing a different aspect.

More about Contexts

A context has the following attributes, that taken together uniquely identify the context:

- **schema** - url of the schema of the data in the context
- **protocols**[] - array of one or more Protocol instances

The kinds of data held by a context depends on the communications protocol (using the term loosely) between the smartwallet and the other party. As will be described next, a Protocol class within the smartwallet represents these data conventions using a schema that is an extension of the Persona schema.

The *DelegatedContext* subclass of Context is described in its own section below.

Protocols

A Protocol class represents a communication protocol used between the smartwallet and an endpoint provided by another party. Each protocol subclass represents a different communications protocol such as SIOPv2, GoogleAccountSync, BasicMessage (DIDComm), etc. Protocol classes have a class method that returns the data schema used when it updates data in that context. These schemas are resolvable from a URL which is written to the *schema* attribute of the Context instance.

A Protocol is an attribute of a Context. Typically, a context has only one Protocol although more than one is sometimes present. Figure 7 shows an example of Alice who has a connection with Santander Bank. This connection has a single context that contains the information that Alice shares with the bank via the OpenID Connect SIOPv2 protocol.

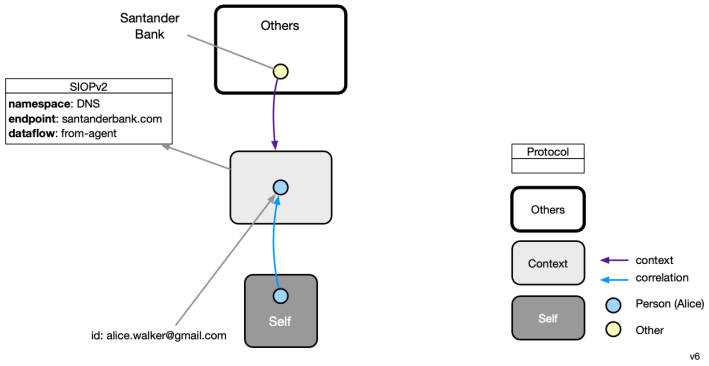

Figure 7: Protocols

Each protocol instance has these attributes:

- **namespace** - a string that indicates the namespace used by the "endpoint" attribute
- **endpoint** - a string identifier that unique identifies the other party with which the person has a relationship within the above namespace attribute
- **dataflow** - one of to-smartwallet, from-smartwallet, bidirectional - indicates the direction of data flow between the smartwallet and the endpoint

Multiple connections

In the example shown in Figure 8, we expand our story about Alice. Alice has organized her connections into two groups. The first represents her role as a journalist, and it contains two connections. The first connection contains a context representing her relationship with Google. The Google context contains her Google account profile which can be updated either using her smartwallet or via the Google website (hence the "bidirectional" dataflow).The second connection contains a context representing her relationship with Twitter. Her Twitter context contains a snapshot of all of her Twitter account information, lists of who she follows, etc.

Her second group, entitled "News" contains one connection comprised of a Person linked to three contexts, all of which are associated with various facets of her relationship to the Olde York Times. The first of these three is the context that she uses, via SIOPv2 to log in to the OYTimes website. The second is a context that contains data her form filler Safari extension uses. The last is a context that establishes a bidirectional connection with the OYTimes using a new (and purely hypothetical for now!) bidirectional data synchronization protocol called MeeTalk. She plays a game for which there is a context (without being within an intervening Group), and she has a direct relationship with her

friend Bob using the DIDComm BasicMessage protocol.

Alice also has two freestanding connections (i.e. connections that are not part of any group). At the far left is her connection to the smartwallet of another person, Bob. In the middle is a connection representing her relationship with a game she likes to play. This connection is also comprised of a single context.

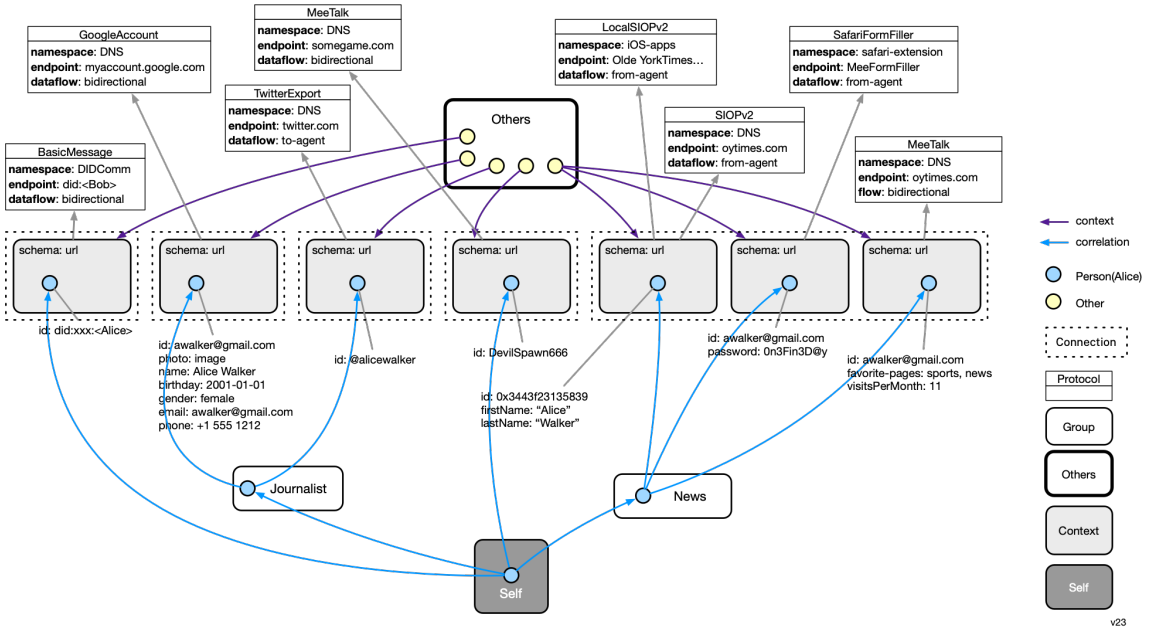

Figure 8: Alice's five connections

A relationship between the smartwallet and another party is called a *connection*. It is represented by one or more other contexts each of which has a protocol (and sometimes more than one). Alice is shown with five connections–one for each of the five Other nodes in her Others container.

Delegated Contexts

Alice takes care of her elderly mother, and helps her mother manage her bank account at Santander Bank. Alice's mother has a smartwallet containing a connection to her bank, the data for which (e.g. her mother's OpenID Connect SIOP claims) is stored in one of the contexts representing this connection. Using her smartwallet, Alice's mother has delegated access to this context to her daughter Alice.

As shown in Figure 9, Alice's mother's connection with her bank is represented by a delegated context. Alice now has the ability to view (and potentially update) information in this context. Information about Alice's mother's account information at the bank might be

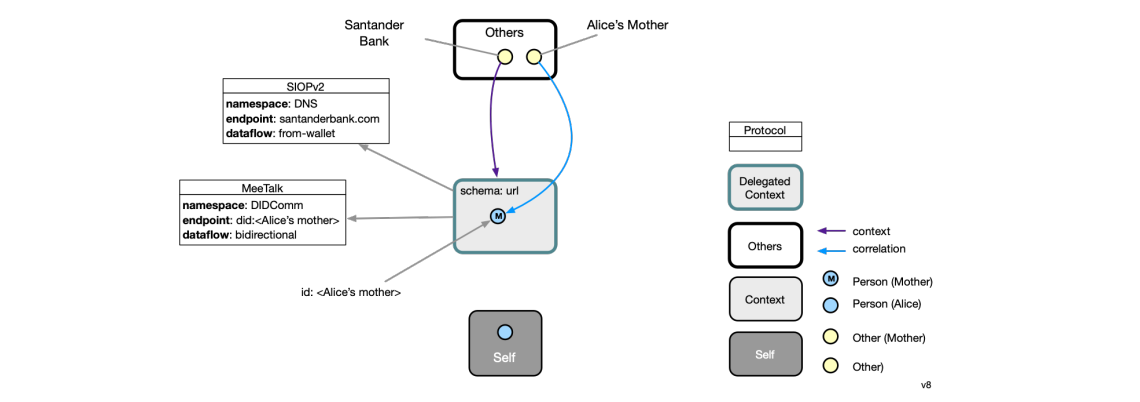

Figure 9: Alice's smartwallet with a connection using a delegated context

helpful for Alice to have while taking care of her mother. Data replication/synchronization is used to ensure that Alice's DelegatedContext is always synchronized with the "original" context on her mother's smartwallet.

Whereas the main point here is giving Alice visibility into her mother's bank, information, it may be possible in some cases for Alice's smartwallet to use this information to authenticate as her mother to the bank (although authenticating as someone else (especially in the case of a bank) is usually a violation of the terms of service of the other party.

5.4.2 Persona classes

Group and context containers contain information about subjects (things) that are described according to the *Persona* schema. In knowledge representation parlance, the Persona schema would be known as an *upper ontology*.

In the Persona schema, people are represented as instances of Person, a *PersonalAccount* class is also defined. These classes are shown below.

Classes

- **Subject** - kind of digital subject about which the smartwallet stores information
- **Person** - a natural person, a subclass of Subject. Each person has the following properties:
  - **claims**[] - a set of zero or more properties. Here are a few examples:
    - givenName
    - familyName

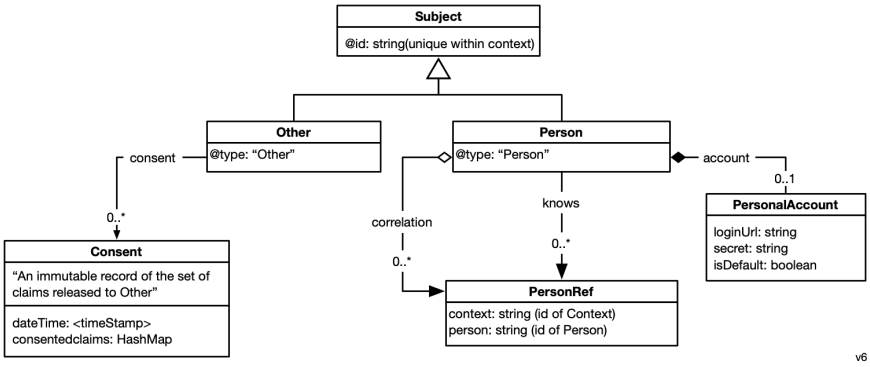

Figure 10: Persona schema

  - phoneticGivenName

- **account** - an optional PersonalAccount at some other party's site or app
- **correlation** - zero or more PersonRefs that act as a link to a target Person object representing another whoness of the link's source's person's selfness.
- **knows** - zero or more PersonRefs that link to a Person representing some other person (other than the smartwallet owner)

- **Other** - a Subject representing another person or a legal entity with which the smartwallet owner has a connection. Each Other object has:
  - **consents** - zero or more Consent objects. Each Consent has:
    - **dateTime** - time stamp of when the person consented to share this set of claims
    - **claims**[] - a set of zero or more claims (note: claim types (e.g. "email address") not their values)

Extensions

Each protocol class will extend the Persona schema by defining Person subclasses, other new object classes and new kinds of relationships. For example the Google Google Account[41] API includes (optional) claims of "name", "gender" and "birthday". The protocol that supports the myaccount API would define these claim types in its schema, and insert a link to this schema in its corresponding context's *schema* attribute.

[41] myaccount.google.com

5.4.3 Datatypes

This section is largely incomplete, but will eventually describe lower level classes that we call *datatypes* that are used by the higher level classes mentioned above. Some datatype classes are shown in Figure 11.

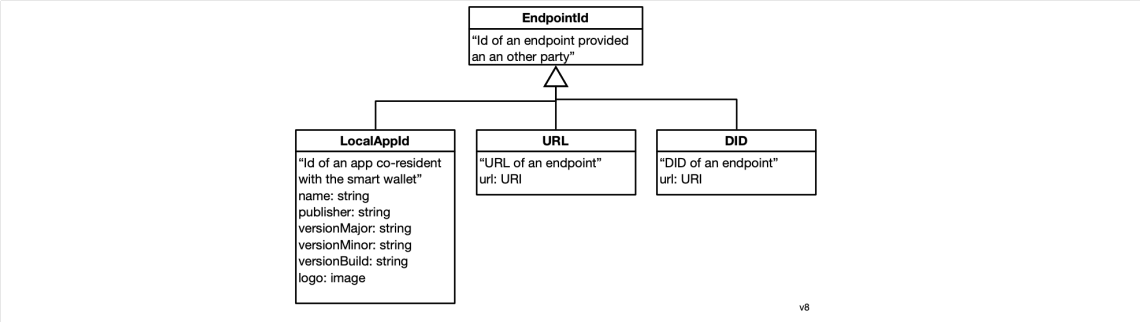

Figure 11: Datatypes

- **EndpointId** - an identifier of an endpoint (e.g. webservice or a local app) supported by an other party.
- **LocalAppId** - A specific kind of EndpointId. Uniquely identifies a service provider's mobile app.

6 Interactions with Apps

We turn now to interactions between a smartwallet and apps.

6.1 Private data sharing

Data held and/or managed by the person's smartwallet and stored on-device, is inherently under the person's control. Data that the person shares with another party or is collected by them in other ways *also* needs to be under their control. Since no technical means exist to control data held by another party, we rely on law. Current privacy laws and regulations are intended to provide this control, but as we've discussed, place such burdens on the person to effectuate their control that in practice this control hardly exists. The solution we proposed is to combine both legal (license agreement) and technical means (smartwallets).

The legal mechanism we propose is the Human Information License (HIL)[42]. The (HIL) is a contract between two parties. The first is the digital service provider. The second is a

[42] docs.google.com/document/d/13aGk5adoncMxxfl5637NfqP6fl6q_op_1CF50UrJNjg

nonprofit, organization called The Mee Foundation (TMF), that represents the community of smartwallet owners. The TMF is a *Mediator of Individual Data* (MID), a term coined by Lanier et al.[13], that enforces the terms of the HIL on behalf of the community.

The HIL imposes obligations on the provider. Among them is the provider's requirement to respect the person's *data rights* to access, correction (editing), and deletion of the information collected and held by them. It covers information that the person may have shared information manually (e.g. by filling in a form, or other kinds of on-app interactions) or shared with them by a person's smartwallet. The HIL requires the provider to implement *data rights* APIs that a smartwallet uses to remotely control this app-held data. In this way, we tie the legal (HIL) and technical means (agents and APIs) together.

The HIL's provisions are intentionally generic. They are designed to meet the needs of the entire community of smartwallet owners. We expect that other contracts containing more specific provisions will be required to meet the needs of more specialized communities. Each community can amend the HIL to meet the specifics they require, provided that they do not weaken the HIL's existing provisions and protections. These specialized communities would organize, govern and operate independent MIDs that enforce their more specialized HIL-based contracts. These specialized MIDs would enter into agreements with one or more providers which would be held to both the generic terms of the HIL as well as the additional, specialized terms.

6.2 App-Agent Interactions

In Figure 12 (a repeat of Figure 4) we show Alice's smartwallet interacting with four apps. At the far left we is an RP Local App running on the same device. On the right we show three other apps. The first is another instance of Alice's smartwallet running on another of her devices. The second is Bob's smartwallet communicating with Alice's smartwallet using the replication protocol common to both agents. The third is an RP's website.

There are four types of intereactions between an app and a smartwallet:

1. **Request** - the app needs information from the smartwallet
2. **App-initiated Sync** - the app has updated information to sync with the smartwallet
3. **Agent-initiated Sync** - the smartwallet has updated information to sync with the app
4. **Delete Context** - the person has directed their smartwallet to delete a connection and all of its associated contexts

Request

When a person uses an app, during the interaction the app may need information about them. The information requested is usually about the person, but could be about other

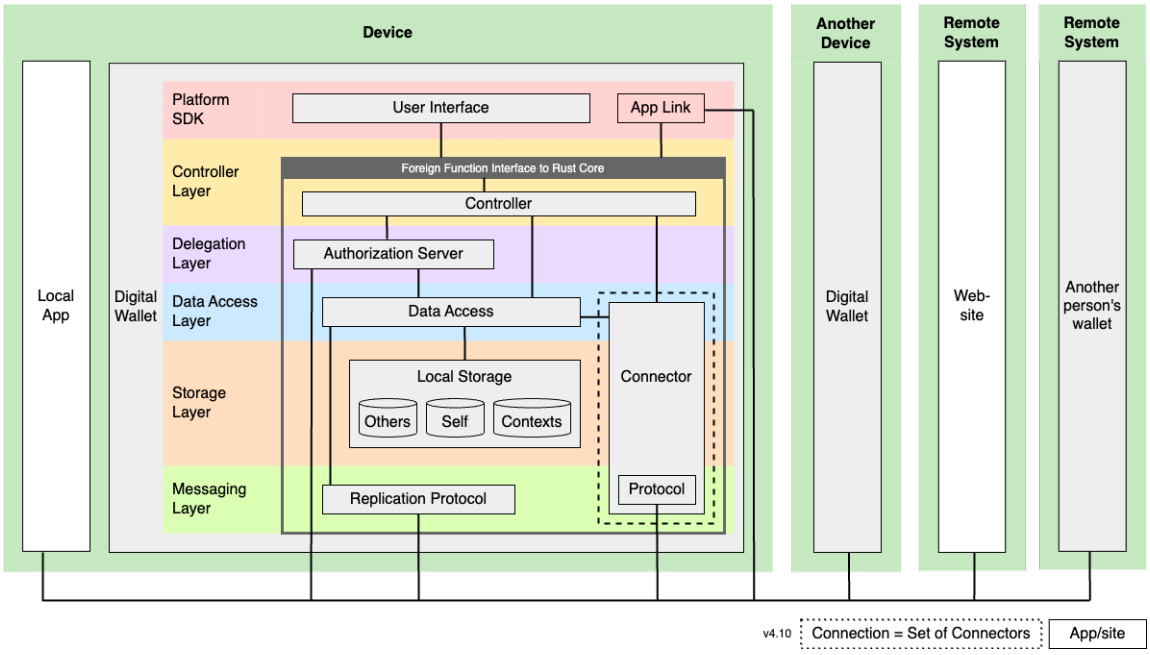

Figure 12: Smartwallet Architecture

people or anything else. The information requested may be a simple list of attributes and values, or something more complicated. The app can express whether each of the requested attributes is required or optional.

Authentication is request wherein the app wants to recognize the person, so it can know if this a first-time or a returning person. Traditionally, an authentication request is implemented using a login/sign-in interaction. However, in a smartwallet-based architecture, a *request* message can be used. To initiate a request the person taps a *Connect-with-Mee* button on the app. This tap initiates an OpenID SIOPv2[43] *Authorization Request*. In the narrow context of authentication the information requested and returned are often called *claims* since they are usually claims made by some entity (possibly the person) about the person. If additional information is needed at any point in the session, the app can again display the Connect-with-Mee button.

Figure 13 shows the *request* interaction in a bit more detail. The smartwallet receives the request message which contains a query describing the kind of information required and/or desired. The smartwallet searches for relevant objects and/or attributes in its storage layer. The smartwallet then performs some or all of the following:

1. **Discuss with person.** Depending on the search results for the information, the

[43] openid.net/specs/openid-connect-self-issued-v2-1_0.html

smartwallet and the person may need to discuss what object and/or attribute values to return. For example, if two conflicting values are found for the same attribute type, the smartwallet may wish to ask the person which (if either) they would like to disclose. If zero or one values are found for a given claim then this step can be skipped, at least for this claim.

2. **Update context.** Populate the context container with the claims (if any) returned from the search.
3. **Display consent screen.** Agent displays consent screen pre-filling what it can and allowing the person to fill in the rest.
4. **Create consent object.** Agent records this consent event.

In the response to the *request* message, the smartwallet returns a non-correlatable *contextId* that can be used for future app-initiated sync operations.

App-Initiated Sync

Apps are required to sync to the smartwallet any changes to claim (attribute) values, as well as any new attribute values. For example, if the person were to use a web form on the app to update and existing or add a new shipping address, then this information must be synced to the smartwallet. This app-initiated sync operation is shown in figure 14.

Agent-Initiated Sync

When the person is interacting with other apps (i.e. apps other than the current one) a new value of an attribute is generated or captured that perhaps should be updated within the current app context. In this case (given appropriate prior consent by the person) an update value of this claim can be automatically synced from the smartwallet to the app. Figure 15 shows this flow. The smartwallet sends one or more sync messages to the app related to the contextId of the current context.

Context Deletion

There is one final app-smartwallet interaction, namely, *context deletion*. This occurs when the user chooses a connection on their smartwallet and taps "Delete." As shown in figure 16 the smartwallet initiates deleteContext messages for each context within the connection.

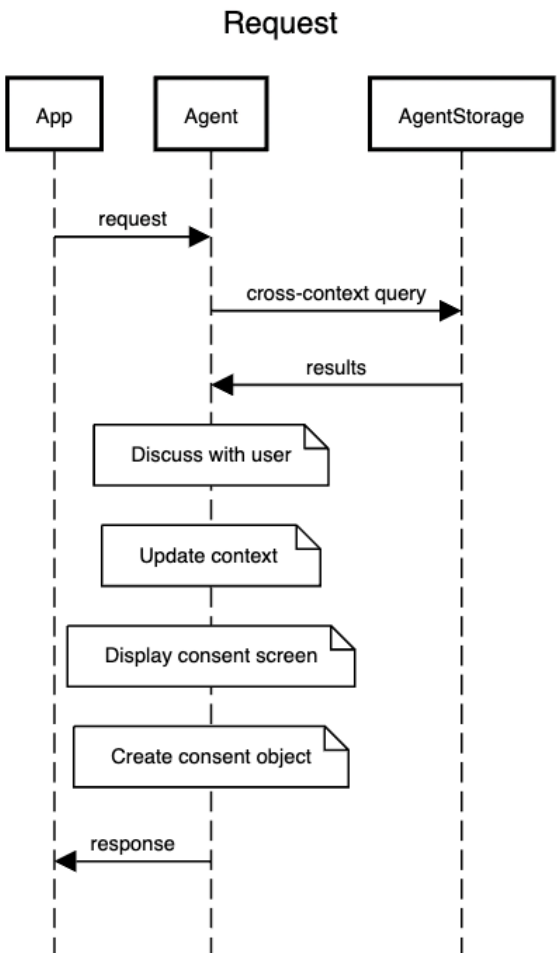

Figure 13: Request

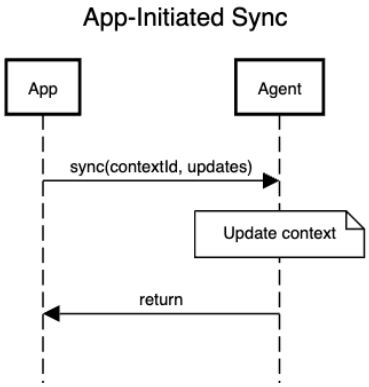

Figure 14: App-Agent Interactions: App-Initiated Sync

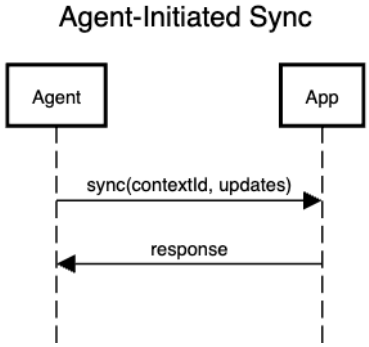

Figure 15: Agent-Initiated Sync

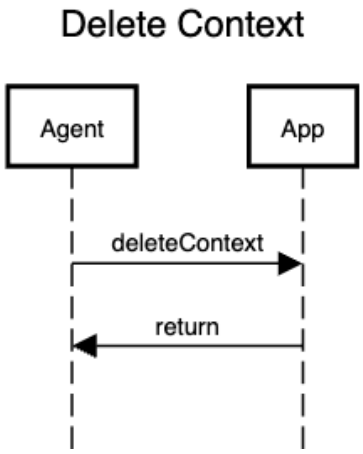

Figure 16: Context Deletion

7 Acknowledgements

Contributors to this paper include Kiril Khalitov, Sergey Kucherenko, Maria Vasuytenko, and Alexander Yuhimenko.

References

[1] Bennett Cyphers and Cory Doctorow. Privacy without monopoly: Data protection and interoperability. *EFF*, 2 2021. URL: https://www.eff.org/wp/interoperability-and-privacy.

[2] Cory Doctorow. Competitive compatibility: let's fix the internet, not the tech giants. *Communications of the ACM*, 64:26–29, 2021. URL: https://dl.acm.org/doi/fullHtml/10.1145/3446789.

[3] Nora A Draper and Joseph Turow. The corporate cultivation of digital resignation. *New media and society*, 21:1824–1839, 2019. URL: https://www.cs.cornell.edu/~shmat/courses/cs5436/draper-turow.pdf.

[4] Anne Josephine Flanagan, Jen King, and Sheila Warren. Redesigning data privacy: Reimagining notice & consent for human technology interaction. *World Economic Forum*, 2020. URL: https://www.weforum.org/reports/redesigning-data-privacy-reimagining-notice-consent-for-humantechnology-interaction.

[5] Gordon Graham. Why the world needs an open source digital wallet right now. *The Open Wallet Foundation*, 2023. URL: https://project.linuxfoundation.org/hubfs/LF%20Research/OpenWallet%20Open%20Digital%20Wallet%20-%20Report.pdf?hsLang=en.

[6] Chris Griswold. Protecting children from social media — national affairs. *National Affairs*, 55:3–17, 2023. URL: https://nationalaffairs.com/publications/detail/protecting-children-from-social-media.

[7] Adrian Gropper. A human rights approach to personal information technology. *Bill of Health*, 4 2022. URL: https://blog.petrieflom.law.harvard.edu/2022/04/12/a-human-rights-approach-to-personal-information-technology/.

[8] Jason I Hong. Teaching the fate community about privacy. *Communications of the ACM*, 66:10–11, 2023. URL: https://dl.acm.org/doi/abs/10.1145/3603718.

[9] Lauren Jackson. A driver's license for the internet. *The New York Times*, 7 2023. URL: https://www.nytimes.com/2023/07/03/briefing/age-verification.html.

[10] Paulius Jurcys, Christopher Donewald, Mark Fenwick, Markus Lampinen, Vytautas Nekrošius, and Andrius Smaliukas. Ownership of user-held data: Why property law

is the right approach. *JOLT*, 2021. URL: https://jolt.law.harvard.edu/digest/ownership-of-user-held-data-why-property-law-is-the-right-approach.

[11] David Kirkpatrick. *The Facebook effect: The inside story of the company that is connecting the world*. Simon and Schuster, 2011.

[12] Martin Kleppmann, Adam Wiggins, Peter Van Hardenberg, and Mark McGranaghan. Local-first software: you own your data, in spite of the cloud. pages 154–178, 2019. URL: https://dl.acm.org/doi/abs/10.1145/3359591.3359737.

[13] Jaron Lanier and E Glen Weyl. A blueprint for a better digital society. *Harvard Business Review*, 26, 2018. URL: http://eliassi.org/lanier_and_weyl_hbr2018.pdf.

[14] Roger McNamee. *Zucked: Waking up to the Facebook catastrophe*. Penguin, 2020.

[15] Friedrich Nietzsche. *The Will to Power*. Knopf Doubleday Publishing Group, 1901.

[16] Eli Pariser. *Eli Pariser: Beware Online" filter Bubbles"*. TED, 2011.

[17] Alex Preukschat and Drummond Reed. *Self-sovereign identity: decentralized digital identity and verifiable credentials*. Simon and Schuster, 2021.

[18] Neil Richards. *Why privacy matters*. Oxford University Press, 2021.

[19] Emma Roth. Online age verification is coming, and privacy is on the chopping block - the verge. *The Verge*, 2023. URL: https://www.theverge.com/23721306/online-age-verification-privacy-laws-child-safety.

[20] Doc Searls. Vrm is me2b. *Project VRM Blog*, 5 2019. URL: http://blogs.harvard.edu/vrm/2019/05/13/me2b-2/.

[21] Daniel J Solove. Introduction: Privacy self-management and the consent dilemma. *Harv. L. Rev.*, 126:1880, 2012. URL: https://scholarship.law.gwu.edu/cgi/viewcontent.cgi?article=2093&context=faculty_publications.

[22] Daniel J Solove. Data is what data does: Regulating use, harm, and risk instead of sensitive data. *Harm, and Risk Instead of Sensitive Data (January 11, 2023)*, 2023. URL: Solove,DanielJ.,DataIsWhatDataDoes:RegulatingBasedonHarmandRiskInsteadofSensitiveData(January11,2023).118NorthwesternUniversityLawReview(Forthcoming),GWULegalStudiesResearchPaperNo.2023-22,GWULawSchoolPublicLawResearchPaperNo.2023-22,AvailableatSSRN:https://ssrn.com/abstract=4322198orhttp://dx.doi.org/10.2139/ssrn.4322198.

[23] Alicia Solow-Niederman. Information privacy and the inference economy. *Nw. UL Rev.*, 117:357, 2022. URL: https://heinonline.org/HOL/LandingPage?handle=hein.journals/illlr117&div=18&id=&page=.