Smartwallets

Paul Trevithick and Sergey Kucherenko, The Mee Foundation

March 3, 2023. Revised August 7, 2025

Abstract

Today's app-centric architecture for personal information has helped fuel the rapid growth of internet apps and sites. Unfortunately and concurrently it has reduced individual autonomy, agency, and privacy. Individuals have no practical means to manage their personal data held by apps/sites. Unfortunately, even if they did, data privacy law has proven inadequate to prevent the disclosure of this information to other app/sites and third parties. We present design considerations for, and the architecture of, *smartwallets*, which can address these issues and restore, at least partially, the power imbalance between individuals and apps/sites.

1 The status quo

1.1 Power

The two main levers of power society are technology and law. Internet technology has resulted in an imbalance between the power concentrated in the hands of a few and the power, or lack thereof, of individuals. Data privacy law, on the other hand, while often based on sound principles has proven insufficient to shift meaningful power and privacy to individuals.

1.1.1 Technology

Apps[1] process personal data in a few different ways: (i) data related to user interactions is stored in *accounts*, (ii) third-party adtech systems track the user and display ads on these apps, and (iii) transaction systems process the user's payment data. We discuss each of these in turn.

Account data. As a user interacts with an app, whatever they type, click, enter, upload is stored in the user's "account." Additional observations, e.g. the kinds of things they click on, and spend time on, are also collected.

[1] We refer to the mobile or local apps, websites, web services, and even other people's digital agents, that a user interacts with as *apps*.

Users have limited power (i.e. control) over their own account data. At best there may be a means to review and update it via an online form. In some cases the app allows the user to download a copy of their account data, although doing so is time-consuming, labor-intensive, and produces dozens of files that the user probably don't know how to use. In some jurisdictions the user has the right to rectify and/or erase account data. Unfortunately, in practice these rights remain almost entirely formal and theoretical due to the unmanageable burden placed on the user to exercise them.

The app provider may sell the user's account data to data brokers[2] 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 form of surveillance of individuals by businesses. Businesses gain ad revenue by implementing surveillance using in-app technologies apps (e.g. third-party cookies, transparent pixels, fingerprinting, etc.) that integrate with hundreds of systems managed by the adtech ecosystem.

Tracking data is behavioral data 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 using. 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[3]. A complex ecosystem of thousands of adtech firms is involved in the supply chain stretching from advertisers, through ad exchanges, to the apps acting in the role of publishers.

Payment data. Apps that sell products or services leverage payment gateways that allow the app provider to receive funds from the user (e.g. via a payment card). In most cases this involves sending financial data (including identifiers) about the user through financial systems run by banks, credit card associations, and their service providers.

In addition to the privacy risks associated with the flow of payment transactions, some app providers also earn money by selling purchase information to data brokers.

Harms. In the data flows just mentioned, the user is relatively powerless over their data. We live today in what Alicia Solow-Niederman calls an "inference economy"[22] wherein big data and machine learning are used to infer traits that form new kinds of personal information–often more sensitive than the underlying source data. Harm and risk can rarely be evaluated outside a specific situation[21], yet it is useful to list representative types of harm. Individuals:

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

- Are vulnerable to data breaches by any of these thousands of apps.
- Have no visibility into what's being gathered, where it's being shared and how it's used.
- Can be spammed by marketers.
- Are vulnerable to identity theft.
- Can be exposed to price discrimination.
- Can be exposed to from hiring discrimination.
- 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."[17] Today, despite significant new regulation, the basic approach to protecting privacy hasn't changed since the 1970s. It is often called *notice and 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.[20]

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

The U.S. population consistently misunderstands the meaning of the term privacy policy.[4] A majority of Americans believe incorrectly the mere presence of a privacy policy indicates a website will not share information without permission.[4]

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

[4] "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."[9]

> 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.[5]

"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."[17]

Progress: GDPR and CPRA

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

In a growing number of jurisdictions, including Europe under GDPR[5] and California under CPRA[6], the person's *data rights*, (e.g. the right to access, rectify 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 renewed global push to protect children's safety through stronger laws and regulations. Although some use other approaches[7], many mandate age verification.[7][10] However, privacy advocates and others have shown that many of the mechanisms for verifying age online weaken anonymity and privacy.[18]

[5] gdpr-info.eu/

[6] thecpra.org/

[7] 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.

1.2 Autonomy

Definition: *freedom from external control or influence; independence.*[8]

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.*"[9] 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, web app, or website, we will simply use the term *app* to refer to all of them.

1.2.2 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."[10]. 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!

1.2.3 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[11], we don't have the ability to do so *peer-to-peer*–i.e. from one person's device to the other person's device. Instead, we're dependent on servers hosted by intermediaries. Further,

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

[9] Phil Windley, personal communication, September 2022

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

[11] berty.tech

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*[12]

1.3.1 Access

Privacy laws such as the GDPR provide the individual the following rights over their personal information regardless of where it is stored:

- Right to rectify
- Right to access
- Right to erasure

These laws are largely not actionable. Without APIs on the service provider's side, as well as personal software to consume them on the individual's side, the burden required to exercise these access rights is unbearable.

1.3.2 Portability

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

In many jurisdictions service providers are required by law to provide individuals with access to their data, but they usually offer this by means of a set of files emailed to the individual as an attachment several hours or days after the request. There are significant problems with implementing portability in this manner. 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 largely unintelligible to them.

Beyond access and export problems, providers generally don't provide "import" APIs to allow the individual to upload their data. Even if an individual could import their data, it first must be transformed into the format of the recipient, since each provider uses their own format. The result is a lack of practical portability.

[12] www.merriam-webster.com/dictionary/agency

Advocacy groups, including the EFF, are pushing for interoperability as an antidote to corporate concentration. This is good, but 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..."[2]

2 Related work

Many initiatives seek to address various subsets of the challenges we've enumerated. 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[3][2].

Work related to re-decentralizing the internet includes: rececentralize.org[13], nlnet.nl[14], DWeb principles[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].

See also 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*[16] movement are squarely aimed at addressing our lack of autonomy.

Particularly relevant is work on personal datastores[25].

Also relevant is work on "local-first" software.[12]

[13] redecentralize.org
[14] nlnet.nl
[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[19] 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)is relevant here. For more about personal datastores see `https://wikipedia.org/wiki/Personal_data_service`

3 Design considerations

In this section, we discuss design considerations for solutions that intend to address the symptoms described in the previous section. We first discuss design considerations related to ensure that smartwallets enforce the user's privacy data rights and are trustworthy. Then, we turn to functional requirements necessary for a smartwallet work with legacy apps as well as new/adapted apps that work within a new trust framework.

3.1 Data rights

In a growing number of jurisdictions new privacy regulations promise *data rights* to access, correct and delete the personal information about users that is managed by app providers. In practice the burden of exercising them across hundreds of apps is unmanageable. Automation on the user's side is required. Smartwallets can provide this automation, although this is only half of the answer. The other half involves constraining how apps use the user's data, and requiring them to implement APIs that the smartwallets can consume.

3.2 Convenience

Convenience is one of the most important design considerations to drive user adoption of smartwallets. To achieve this a smartwallet must *automate* tasks related to controlling and managing a person's information.

3.3 Inclusivity

Smartwallets must be affordable by all socio-economic classes, not just those better off. Thus, solutions that incur monthly hosting costs to the user are disqualified. Smartwallets should be available at no charge and run on a device the user already owns, although there may be additional costs incurred for any additional storage required for large on-device datastores.

3.4 Loyalty to the user

Much of the power asymmetry described in the first section is created by the economic incentives for online service providers to monetize user's personal information. Providers do just enough in the user's interest to ensure that they keep using the service, so they can capture personal data to monetize. Personal data, after all, is now considered to be an asset class, and the more of it that's collected the better. For a user to trust that their smartwallet works *exclusively* on their behalf, the *smartwallet provider* (i.e. organization that develops and provides it) must not have an economic incentive to be disloyal to the user. This can be ensured by designing smartwallets such that the personal information that they process is never accessible to the smartwallet provider. This largely eliminates

the need to trust the smartwallet provider organization, their security infrastructure, their processes, etc.

3.5 Open source

The transparency of open-source software builds confidence that any smartwallet built from this source code is trustworthy. In open-source software the source code is visible to anyone to review and audit to ensure that the smartwallet is secure, free from vulnerabilities, works in the person's interest, and performs as expected.

3.6 Trustworthy smartwallets

Users need to trust that the smartwallet developer/provider organization is itself trustworthy. Their financial incentives should be aligned with the user's interests. Nonprofit or similar organization structures can increase trust that smartwallets offered by them have no financial incentive to exploit the user's data for their own advantage.

3.7 Trust Framework

Once data is shared from a smartwallet to an app there are no technical means available to constrain what the recipient app can do with it. No technical means, for example, can prevent them from selling it others. Instead, legal means (trust frameworks) must be employed.

Some apps will be willing to join a trust framework and *license* the user's information received from the user's smartwallet. If so, they would agree to the terms of a license agreement which include terms which respect the person's privacy rights. This contract is signed by a trusted organization[26] that represents the community of smartwallet users thereby making the processes effortless for them. Lastly, this organization is responsible for enforcement of the contract's terms, again, on behalf of the user.

3.8 Human-centricity

Many of the challenges described in Section 1, The status quo, 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

[26] 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.

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 unbearable.

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.

3.9 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. Facebook) to have a single identity[27], 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, but 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."[17, 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[28] Kim Cameron[29] introduced two useful definitions based on archaic English:

- **Selfness**: The sameness of a person or thing at all times or in all circumstances. The

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

condition of being a single individual. The fact that a person or thing is itself and not something else. Individuality, personality.

- **Whoness**: 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.

Figure 1: Multiple whoness-contexts around a single selfness

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.

3.10 Local-first

In this section we argue that a local-first architecture is preferred.

A user's personal datastore may be on-device or in the cloud. By *on-device* we mean that the individual's datastore and processing is on their own phones, laptops, and/or home servers. By *cloud-based* we mean that the person's datastore and processing lives in the

cloud (e.g. on a SOLID[30] pod). The local-first software[31] principles are highly relevant. Although there are other points of view, we contend that if many people are using the solution, 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 nature aggregates large numbers of personal datastores at one cloud service provider location, and thereby creates a much larger economic incentive for hackers.

3.10.1 Synchronization

A user may have two or more smartwallets running on multiple devices each of which is only intermittently connected to the internet. The user's data needs to be kept consistent across these smartwallets and devices, at least eventually. This requires that the person's smartwallets implement data replication and syncing between themselves in a peer-to-peer (P2P) fashion. Unfortunately pure P2P communication between smartwallets running on differing device platforms remains an unsolved problem and intermediate relay servers are sometimes 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 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 system or smartwallet to come online.

3.10.2 Backup

One disadvantage of *noncustodial*, on-device architectures (as compared to cloud-based architectures) is the vulnerability that smartwallet users who are not diligent about backing up their devices (e.g. to an online service) face of losing their smartwallet-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.

Smartwallets could implement backup/restore approaches that would provide recovery from even this disaster, however these approaches are complex and problematic. The smartwal-

[30] solidproject.org
[31] www.inkandswitch.com/local-first/

let's data must be 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[32], and social recovery have been proposed, although this remains an area of active research.

3.11 Delegation

In *A Human Rights Approach to Personal Information Technology*[8] 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])

He 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[33] at the 2022 Identiverse conference provides additional detail (see slides[34]). It explains that only a decentralized architecture can implement the Gropper 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.

4 Smartwallets

Smartwallets are a solution to the problems described in the first section and take into account the design considerations in the second section. Through a combination of technical and legal mechanisms a smartwallet gives individuals control over their personal information as they interact with websites, mobile apps, as well as other people's smartwallets. The solution combines a legal contract with a trusted, personal agent[35] integrated with a traditional digital wallet[6].

[32] en.wikipedia.org/wiki/Shamir%27s_secret_sharing
[33] identiverse.com/idv2022/session/841489/
[34] drive.google.com/file/d/1lwaMVkG4kLi7z6cXhqMx-DGkUww9azW3/view
[35] Similar ideas have been proposed by others. See *personal user agents*, in [5, p24]

A smartwallet is a native software application (e.g., written in Swift on iOS, Kotlin on Android, etc.) that runs on a user's devices (e.g., mobile phone, laptop, etc.). It maintains a local, private datastore of the user's personal information.

Apps can request personal information from, and provide information to, the user's smartwallet using the MDN protocols (see section 4.1). The information may flow between the app and the smartwallet in one direction, the other, or in both. The app reads and writes data using the smartwallet's data model. This data may include both structured and unstructured data and may include digitally signed documents (e.g. Verifiable Credentials, etc.).

4.1 Mee Data Network

The Mee Data Network (MDN) is a set of protocols designed to support data sharing between MDN nodes. These nodes may be integrated with a provider's apps or websites, and/or within the user's smartwallet if they choose to install and configure one or more of them on their devices.

The MDN includes a trust framework wherein apps authorized by The Mee Foundation implement MDN protocols and to agree to the terms of the MDN License. This license requires that apps abide by certain privacy principles regarding how they handle the individual's data (e.g. requiring explicit consent for collection, processing, storage and sharing of the person's data) as well as implement the MDN protocols. These protocols, combined with the license, enable *private sharing* between the smartwallet and the app or site.

Private sharing allows the user to share personal information with confidence that it remains under their control. Following intellectual property law precedents, the user licenses their information to the app rather than transferring a copy of it in the hope that the app will treat it with care. Using their smartwallet, the user can exercise their rights to access, correct and delete their information stored by the app.

The MDN is described in more detail in section 6.

4.2 Benefits for the individual

4.2.1 Privacy

A smartwallet increases the user's privacy as follows:

- If the smartwallet includes a browser extension component, it can add the Global Privacy Control[36] field in the HTTP header expressing the user's privacy preference. The extension may also include the ability to delete third-party cookies and other kinds of trackers used in surveillance advertising. Smartwallets can participate in

[36] globalprivacycontrol.org

new, *private advertising* networks, that don't rely on cookies, trackers, data brokers, etc. but instead rely on user profiles that are anonymized, and never shared outside the ad network.

- If the smartwallet interacts with apps that are part of the MDN, these apps agree to process the user's personal information under the terms of MDN License. By default, the app can't sell, transfer or share the user's information without their consent.

4.2.2 Protection of Minors

Minors can be given a special child-oriented smartwallet by their guardians. This kind of smartwallet is an enabler to provide the minor 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 smartwallet an age verification credential. When the minor uses a first-party app, the smartwallet can signal that the minor wishes to have an age appropriate experience. In response the app can request the age verification credential from the minor's smartwallet and adapt its experience accordingly.

4.2.3 Autonomy

Smartwallets provide individuals with digital embodiment of themselves. Over time, the smartwallet develops rich, context-specific data profiles about them. This embodiment can move autonomously under the user's control between first-party apps. A *local* smartwallet can do so independent of any external administrative authority.

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 usage of smartwallets grows, surveillance free, end-to-end encrypted communications interconnect these users. These communications can be designed with minimal reliance on cloud-based relay servers which are often needed to buffer messages to endpoints that are temporarily offline.

4.2.4 Agency

A foundation for Personal AI

Rather than requiring individuals to trust a shared AI-in-the-cloud service with all of their sensitive personal information, a better approach is to have the *Personal AI* algorithms run on the individual's devices. These algorithms read and write personal information to/from the person's smartwallet.[37]

[37] Iron Man's J.A.R.V.I.S.[38] is an example of this architecture and offered Iron Man complete privacy.

Logging in without passwords

Smartwallets enable the user to log in to apps using a variety of password-less authentication technologies. The smartwallet knows who the user is because the user authenticates to it, so the smartwallet can represent the user in their interactions with apps, and can do so without revealing correlatable identifiers. This is both private and convenient.

Wielding credentials

In real life an individual can, say, present their driver's license to a wine seller to prove that they are of drinking age because the wine seller trusts the license issuer. This interaction is privacy-respecting because the presentation interaction is never disclosed to the issuer. This driver's license use-case involves the individual *wielding* a trust credential. Unfortunately, there is no commonplace way to do this online. There's no standard way to be issued a credential, hold it in a digital agent (acting as 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 an individual to prove something one party states about them, to another party. Digital wallets are emerging to meet this need and this wallet-like capability is included in a smartwallet.

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 about themselves. Individuals must manually re-enter personal information into each app, endlessly repeating themselves. They lack an agent that can automatically present information on their behalf.[39]

This endless repetition is a symptom of the internet's silo-ed architecture wherein each app maintains its own database of personal information. The individual has the hassle of repeated data entry, and the app offers a less-than-optimal user experience. With a smartwallet the user no longer has to repeat themselves as they move from app to app.

This inability to present ourselves digitally is a contributing factor to the concentration of corporate power 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. Continuing with the shopping example, smartwallets can represent an individual to any e-commerce website, and thereby provide the same Amazon-like, frictionless user experience that can mitigate corporate concentration and "natural" monopolies.

Infer and present ad profiles

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

A smartwallet can generate on-device an ad profile by inferring traits from an individual's browsing behavior. The smartwallet's user can review and edit this profile, and may choose to share it with apps that are supported by interest-based *private* advertising technology. 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[40].

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. At present, there is no standardized way to do this online. This is especially problematic in healthcare scenarios where a healthcare provider needs access to health-related data about a patient, but the patient is not in a situation where they are able to provide it by themselves and must instead rely on someone else, e.g. a family member to grant the needed permission. In the online world each service provider not only possesses the individual's data, but they manage it in such a way that it is impossible for the individual to delegate rights to it to others.

Content filtering

Social networking platforms have replaced human content editors with algorithmic filters. Individuals may think that they are seeing a balance of content whereas in reality they are trapped in what Pariser called "filter bubbles."[15] 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] Smartwallets can achieve this.

Account management

The individual carries 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 individual to spend more time at sites that already have their information. The relative convenience of shopping on Amazon vs. other e-commerce sites is partly a consequence of the individual not having an easy way to manage and update their personal information at multiple sites–it's just easier to buy things on Amazon because Amazon already has all of their personal information.

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

5 Smartwallet implementation

A smartwallet sits at the center of an architecture wherein the user's interactions with apps radiate out from it. "When we put the user at the center, and make them the point of integration, the entire system becomes simpler, more robust, more scalable, and more useful."[1]

This human-centered architecture necessitates that a smartwallet must be able to interact with a wide variety of different kinds of apps. Consider the user's interactions with six apps. The first app might want to know the user's email address, and it might ask for it in a web form. A second form-filler app (which might be a browser extension integrated with the smartwallet) uses this value to fill in the form. A third app might support password-less sign-in (e.g. using OpenID Connect) that leverages a smartwallet acting as the so-called *identity provider*. A fourth might request a digital driver's license credential from a fifth digital wallet app which can present this credential–a credential that had presumably been installed into it from a sixth credential-issuing app.

5.1 Self and Contexts

The smartwallet represents both the person's single *selfness* and a set of *whonesses*, each used in the context of their interaction with a different app.

The selfness of the person is represented by a person entity in a data container called the *self*. The person entity in the self is the point of integration across contexts each of which may use differing identifier namespaces, protocols for communication, and data schemas. The contents of the self are holistic and therefore quite sensitive and would normally not be shared with others.

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 smartwallet user knows that each of these separate contexts contain representations of them. Each context represents an interaction with an external app.

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

In our example Alice has four digital relationships called *connections*. Each of the four connections shown in the diagram has only a single context, although in general a connection may be represented by more than one context. Since, as will be described more fully later, a context is associated with exactly one communications protocol, in cases where N

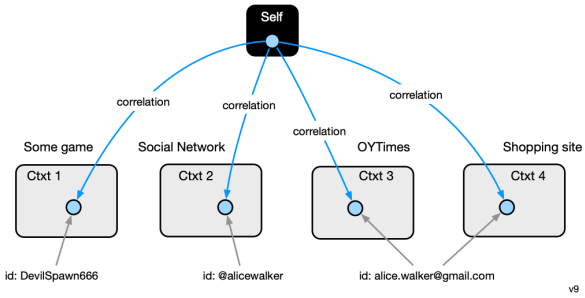

Figure 2: Alice in four contexts

communications protocols are used in a single connection, the connection will contain N contexts.

5.2 Functionality

Figure 3 summarizes the functionality of a smartwallet. The first set of rows list the set basic smartwallet functions. The cells in light and dark green show the progress of implementation work being led by The Mee Foundation.

Smartwallet Functionality		iOS	Android	Mac/ Win/ Linux
Connection Data Management Functions	**Request** access to a context managed by another app			
	Grant access to a context managed by the user			
	Organize relationships into a set of connections and contexts			
	Sync contexts across MDN nodes running on user's devices			
	Delete connection			
	Consent to share data with an app			
	Edit data in self-asserted contexts			
	View data in connection's context			
Other	**Authenticate** the user			

Feasible | Under Development | Implemented

v37

Figure 3: Smartwallet Functionality

5.2.1 Functionality

In addition to authenticating the user to the smartwallet, a smartwallet performs the following functions related to the management of the user's connections :

- **Organize** the relationships the user has with apps into a set of connections and contexts.
- **Request** access to a context managed by another app.
- **Grant** access to a context managed by the user.
- **Sync** contexts across user's devices.
- **Delete** all data associated with this set of contexts.
- **Consent** to share data with an app.
- **Edit** data in self-asserted contexts within a connection.
- **View** data in a context (connection).
- **Authenticate** the smartwallet user.

5.3 Smartwallet architecture

The architecture of a smartwallet is shown in the center of Figure 4. The user must have at least one smartwallet running on a mobile phone. They may choose to have additional smartwallets running on other devices. The state of the smartwallets is replicated and synchronized across this pool of smartwallets.

Terms such as *self*, *context*, and *connection*, used in the following are described in section 5.5 where the Persona data model is described.

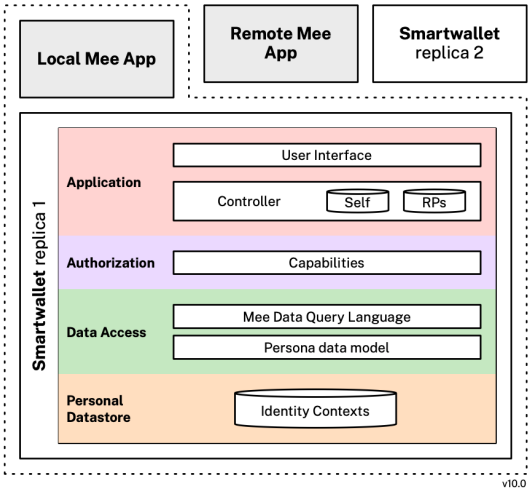

Figure 4: Smartwallet Architecture

5.3.1 Application layer

The Application layer of a smartwallet consists of a User Interface (UI) subcomponent and the business logic in the Controller subcomponent.

The UI provides an interface that enables the user to manage their data sharing relationships with apps. Using this UI the user can add and delete connections. Within each connection they can consent to data being shared from their smartwallet, see what data is involved in the connection, and in some cases edit attribute values.

The Controller handles requests from the UI. It manages the Self and the RPs data containers. Controller is also responsible for replicating and synchronizing the state of the user's smartwallets (e.g. using the Willow Protocol[41]). Note that since the storage capacity of each smartwallet node may vary based on the storage capabilities of its host device, whereas all Self and RPs data is always replicated on each node, some Contexts may be omitted on the nodes running on storage-constrained devices.

The Controller, using the Data Access layer, updates the attributes of Person objects in contexts. It is responsible for creating and deleting connections.

5.3.2 Authorization layer

The Authorization Layer manages the granting, verification and revocation of capabilities.

5.3.3 Data access layer

The Mee Data Query Language subcomponent is responsible for management of the user's data whether it is stored locally, replicated on another of the user's smartwallets, or managed by a service provider's app. It exposes data contexts via the Controller to the User Interface where it can be viewed and in some cases edited.

5.3.4 Personal datastore layer

This layer manages local data contexts (some of which may be accessed by Connectors). Data is encrypted using FIPS-compatible algorithms.

5.4 Data sharing

A smartwallet contains an embedded personal datastore, although storage of the user's data may also be distributed among multiple MDN nodes within MDN-compatible apps.

[41] willlowprotocol.org

5.5 Persona data model

This section describes the data model used by smartwallets to represent personal information. The user's data may be replicated across multiple smartwallets on different devices, but we focus here on the logical model, not these replicas. The data model can be thought of as a two level hierarchy of data containers each of which holds *Person* instances representing the user. The top layer consists of a single *Self* container. The bottom layer consists of *Context* containers.

These Person instances are connected into a directed graph that spans these 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. 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.

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 four connections, each to one of four apps: a game, a social network, the Olde York Times and a shopping site. Each of these connections is represented by a single context, although more complex connections may include more than one context. The whoness, i.e. the aspect of Alice that she exposes in each context, is represented by a Person node in each.

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 four of these other "relying parties" to Figure 5, and added a new kind of container, called RPs, to contain them.

The personal information flowing through the connections 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 using an app, or sensed by a local app's sensor, or generated by the other party based on direct on-site or on-app interactions with the person.

5.5.1 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.

Classes

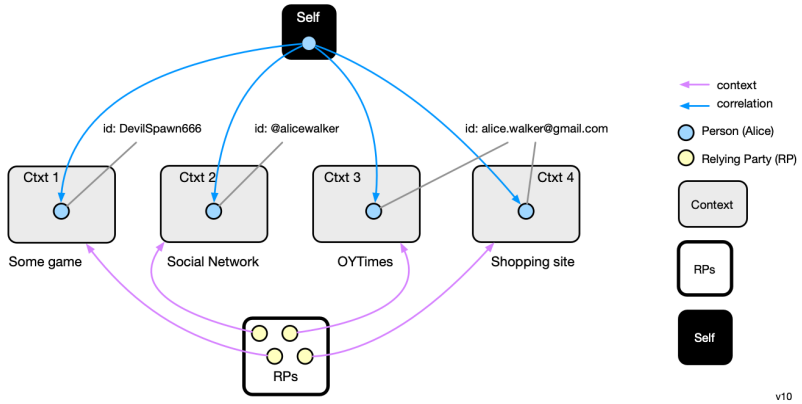

Figure 5: Alice's Self and Relying Parties

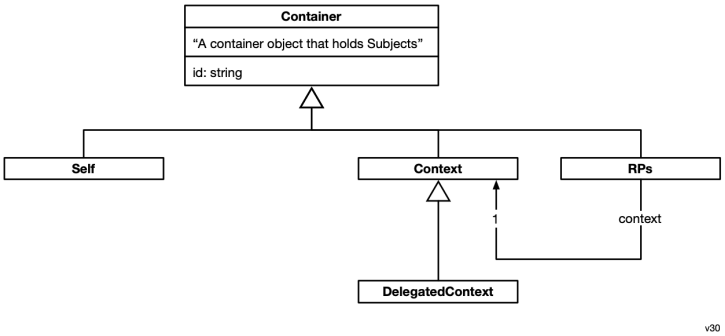

Figure 6: Container classes

- **RPs** - a container holding a set of Relying Party (RP) nodes (see Smartwallet Classes). Each RP node represents a party with which the person has a connection. These RPs may be other people or corporations, such as a digital service provider. Each RP has the following properties:
  - **Context** - a single Context that captures one aspect of the connection between the user and some RP.
- **Self** - the single data Container holding a single Person node that represents the selfness of the user.
  - **id** - user identifier (e.g. email)
- **Context** - a Container holding a Person node that represents the user in a specific aspect of their relationship (called a *Connection*) with some RP.

Multiple connections

In the example shown in Figure 7, we expand our story about Alice. Alice has five connections contextualizing her relationships with each of five organizations and/or people. We discuss each connection moving left to right in the diagram:

- **Bob**. Alice has a connection to Bob mediated by an app that uses the DIDComm BasicMessage protocol[42].
- **Some game**. Alice has a connection with a game she likes to play. It contains a context representing this game. She uses id "DevilSpawn666" as her identifier in this context.
- **X**. Alice has a connection to X social network. It contains a context representing her X account. Her id is "@alicewalker" on X.
- **Google Account**. Alice has a connection mediated by an app that accesses her Google account. The context for this app contains the attributes of her Google account. This context uses her "awalker@gmail.com" id.
- **Olde York Times**. Alice has a connection to the Olde Yorke Times (hypothetical) news media site. The context captures her sign-on relationship using OpenID SIOPv2 protocol using her id, "0x3443f23135839". The context holds information she has entered using a form filler app as well as her account information managed by this site.

A relationship between the smartwallet and another party is called a *connection*. It is represented by one or more other contexts. Alice is shown with five connections–one for each of the five RP nodes in her RPs container.

Delegated Contexts

Alice takes care of her elderly mother, and helps her mother manage her bank account at Santander Bank. Her 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 8, 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 her mother's account information at the bank might be 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.

[42] didcomm.org/basicmessage/1.0/

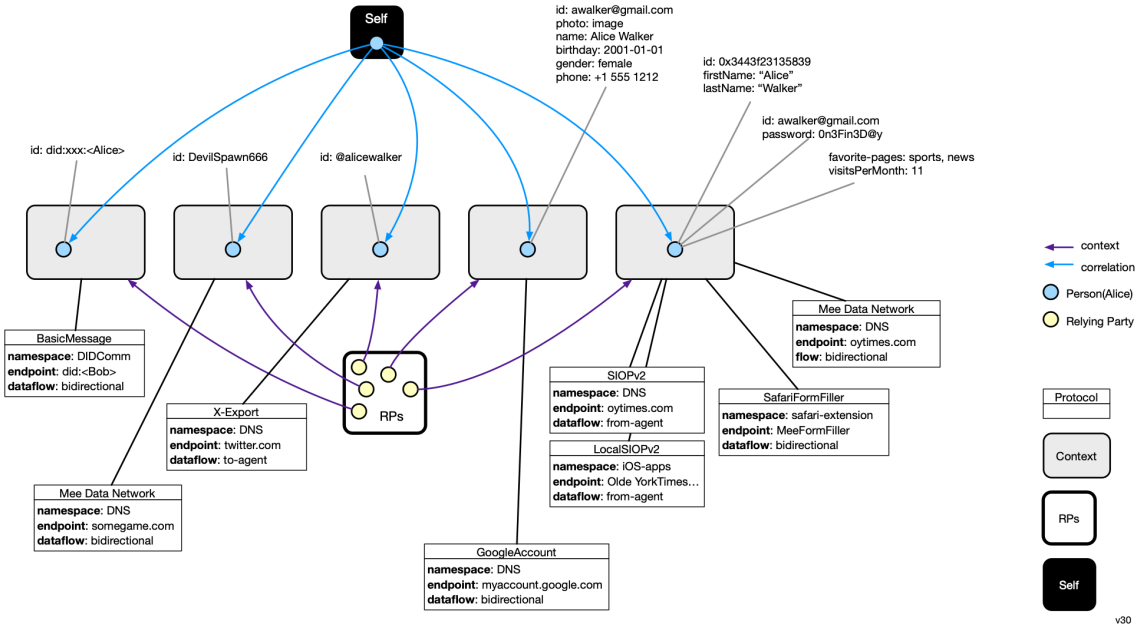

Figure 7: Alice's five connections

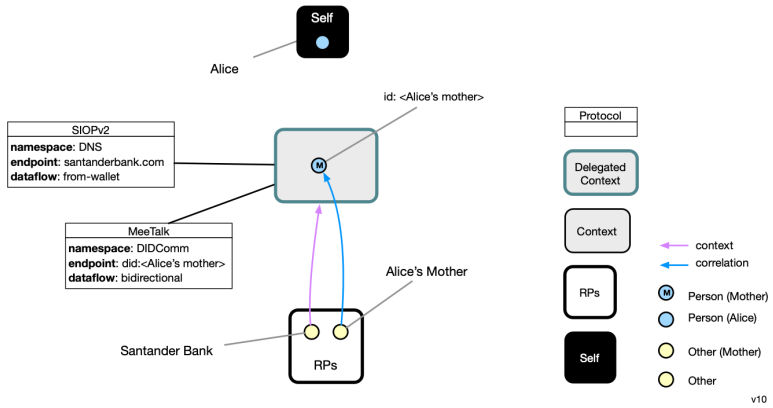

Figure 8: Alice's smartwallet with a connection using a delegated context

5.5.2 Smartwallet 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.

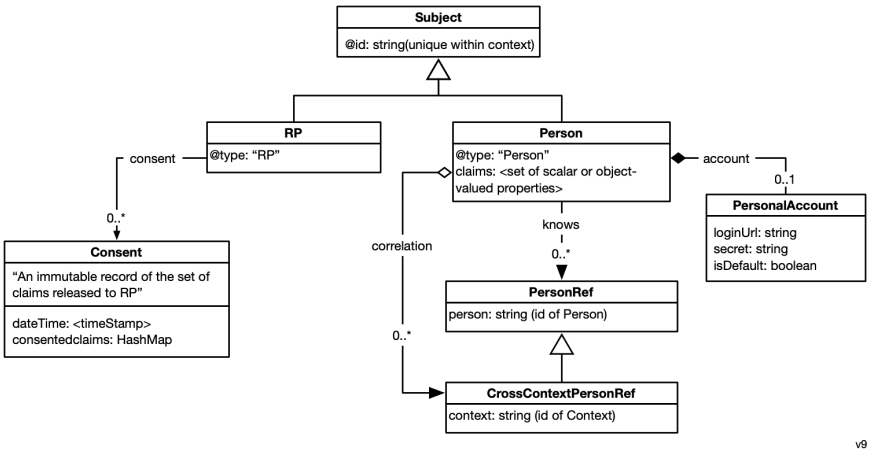

Figure 9: Persona schema

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. These properties may be structured (e.g. a physical address (e.g. from vCard)) or scalars. Here are a few examples of scalar *claim* properties:
    - givenName
    - familyName
    - phoneticGivenName
  - **account** - an optional PersonalAccount at some other party's site or app
  - **correlation** - zero or more CrossContextPersonRefs each of which acts as a link to a target Person object in another Context. Both the source Person and the

target Person can be thought of as contextualizations of the same underlying person.

  - **knows** - zero or more PersonRefs that link to a Person representing some other person (other than the smartwallet user) in the same context

- **RP** - a Subject representing another person or a legal entity with which the smartwallet user has a connection. Each RP object has:
  - **consents** - zero or more Consent objects. Each Consent has:
    - **dateTime** - time stamp of when the user 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)

6 Mee Data Network

Although smartwallets can form connections with many kinds of existing apps using a variety of protocols, we describe here a network of apps and smartwallets called the Mee Data Network that use a specific set of protocols and adhere to a specific trust framework. These two, taken together, offer smartwallet users particularly strong privacy guarantees.

6.1 Private data sharing and the Mee Data Network

It is obvious that data held and/or managed by a user's smartwallet and stored locally on a device the user owns, is inherently under this user's control. The challenge is that data that a user shares with another party or that is collected by that party in other ways *also* needs to be under the user's control. Unfortunately, it is impossible using solely technical means to remotely control data held by another party. Privacy laws and regulations on the other hand, while intended to provide this control, in practice place such burdens on the user to effectuate this control that it hardly exists. The solution is to combine both legal (license agreements) and technical means (smartwallets and apps on the Mee Data Network).

The legal mechanism we propose is the Mee Data Network License (MDNL)[43]. The MDNL is a pairwise contract between two parties. The first is the service provider providing an app. The second is an organization that represents the community of smartwallet users (e.g. The Mee Foundation). This organization acts as a *Mediator of Individual Data* (MID), a term coined by Lanier et al.[13], that enforces the terms of the MDNL on behalf of the community.

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

The MNDL imposes obligations on the app provider, among which is the requirement to respect the user's *data rights* to access, correction (editing), and deletion of the information collected and held by them. The MNDL covers information that the user may have shared 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 MNDL 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 (MDNL) and technical means (smartwallets and APIs) together.

The MDNL's provisions are intentionally generic. They are designed to meet the needs of the entire community of smartwallet users. 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 MDNL to meet the specifics they require, provided that they do not weaken the MDNL's existing provisions and protections. These specialized communities would organize, govern and operate independent MIDs that enforce their more specialized MDNL-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 MDNL and the additional, specialized terms.

7 Acknowledgements

Contributors to this paper include Kirill Khalitov, Alexander Yuhimenko, Maria Vasuytenko, Vlad Fisher, and Xenya Shatalova.

References

[1] Joe Andrieu. Vrm: The user as point of integration. *joeandrieu.com*, 6 2007. URL: `https://blog.joeandrieu.com/2007/06/14/vrm-the-user-as-point-of-integration/`.

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

[3] 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`.

[4] 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`.

[5] 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.

[6] 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.

[7] 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.

[8] 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/.

[9] 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.

[10] 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.

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

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

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

[18] 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.

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

[20] 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.

[21] 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.

[22] 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=.