Identity Agents

Paul Trevithick, The Mee Foundation

March 18, 2023

1 Power Asymmetry

While the internet has brought new services and experience to billions of users, it has also resulted in a power asymmetry between the digital service providers and their users regarding these users' digital identities and personal data. These providers have accumulated power relative to their users.

The internet's designers endeavored to create decentralized architectures that pushed computation and storage to the edge and minimize centralized control. However, in the last couple decades economic factors and so-called natural monopolies, preferential attachment network effects, economies of scale, and the relative ease of creating centralized solutions have all contributed to concentrations of power on the provider side.

When Berners-Lee created the web, it was a decentralized platform. Anyone could publish a website and link to any other side. But as the web has grown from an obscure research-sharing community into a global medium for commerce, communication, journalism and entertainment, the power dynamics have shifted. Today, huge companies like Amazon, Meta, Google, and Netflix dominate the web. These corporate giants enjoy an enormous amount of control not only over what people see and do online but over users' private data.[1]

In response, many initiatives and projects are propose alternative approaches and technologies. Here are just a few examples: rececentralize.org[1], DWeb principles[2], The Web3 Foundation[3], the Decentralized Identity Foundation(DIF)[4], "local-first" software princi-

[1] redecentralize.org
[2] getdweb.net/principles/
[3] web3.foundation/
[4] identity.foundation

ples[5], ProjectVRM[6], Blue Sky[7], and the Decentralized Information Group[8].

To reinforce our focus on power relationships, we use the term *computational power* to refer to software tools that work (i.e. provide agency) "on the user's side" for, and *exclusively* on behalf of, the user. We are not talking about whether the average user's phones and laptops have sufficient computational power; they do. We're saying that this raw computational power and storage is not leveraged by tools that push back on app provider's power, and engage digitally and automatically to empower the user to better manage their relationships with providers on terms more beneficial to them. Since our discussion applies equally provider's mobile apps, webapps, and websites, we simply use the term *app* to refer to all of them.

Power asymmetry lies at the root of a diverse set of related symptoms, most of which erode privacy, since privacy and power are highly interrelated concepts[9]. These privacy eroding symptoms include a lack of personal autonomy, a lack of personal agency, and third-party surveillance.

1.1 Lack of Autonomy

au•ton•o•my: *freedom from external control or influence; independence.*[10]

Independence. In the physical world each of us is a separate, independent entity. Each of us has a self that embodies our individuality. We "bring" that independent selfness to interactions with others, with vendors, etc. while understanding that this independence is not absolute–we are still to some extent dependent on common, shared systems, laws, environments, and so on. By contrast, online it has been said that "we have no *digital embodiment*."[11] Our identities are provided to us by digital service providers (e.g. in the form of a Facebook identity, or an Amazon account). Without them we don't exist. Anyone who has been banned from a platform, or uses a platform that is shut down is sharply reminded that their digital identity exists at the pleasure of that platform. Our provisional existence is the original power asymmetry. Efforts create personal datastores, or even more to the point, those that strive to provide each of us a *self-sovereign identity*[6] are squarely aimed at addressing this issue–the word "sovereignty" certainly evokes power.

Ownership. Our personal data is collected and held by businesses (first-parties) as we interact with their apps, not by us. This pattern *app-held data* is so common that it's hard

[5] inkandswitch.com/local-first/
[6] blogs.harvard.edu/vrm
[7] blueskyweb.xyz/
[8] dig.csail.mit.edu
[9] Consider the title of Véliz's recent book, "Privacy is Power"[7]
[10] languages.oup.com/google-dictionary-en/
[11] Phil Windley, personal communication, Sepember 2022

to imagine an alternative. Our data is not free from external control by apps, because it is generally stored and managed by them.

Our data is also collected and held by third-parties (e.g. data brokers) with whom we have no interactions at all. In short, it's been said that "everybody has our data ... except us."[12].

As we'll discuss more later on, there are alternative approaches. One is "user-held" data[2], where your data is held by you in a personal datastore)[13]. Another is following "local-first" software principles.[4]

Lock-in. As we've just mentioned our online existence is provisional. Further, this existence is bound to the provider from which it originated. Providers hold our data, and although in many jurisdictions providers are required to allow us to have access (e.g. to request that we can download a copy), we lack the technical means to accept the data stream and hold (e.g. in a personal datastore) that has the ability to subsequently transform it into other formats and schemas so as to make it reusable in other contexts. This lack of agency results in our data being held hostage, i.e. without autonomy.

Peer-to-peer With a few exceptions, e.g. technologies like Berty[14], internet users, when they communicate person-to-person don't have the ability to do so *peer-to-peer* from their edge devices to the other person's device. Instead, they are dependent on servers hosted by intermediaries. Whereas these days the messages themselves are end-to-end encrypted, the metadata (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's server.

1.2 Lack of agency

a•gen•cy: *the capacity, condition, or state of acting or of exerting power* [15]

Wielding credentials. In the offline world you can autonomously present your drivers license to a wine seller in order to prove that you are of drinking age since the wine seller trusts the license issuer. The interaction is privacy-respecting because the presentation interaction is not disclosed to the issuer. This could be described as "wielding" a trust credential. At present, there is no equivalent way to do this online. There's no standard way to be issued a credential, hold that credential in digital wallet, and then present that credential. With a few, domain-specific exceptions (e.g. cryptocurrency), there is no common, generalized

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

[13] 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`

[14] berty.tech

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

method for you to prove something about yourself as stated by one party about you, to another party online.

Data presentation. One reason for form filling and other kinds of data entry on providers apps/sites is that the user, even if they were equipped with a personal datastore, lacks the ability to present personal information digitally to the provider. Instead the information must be re-entered manually at each provider. The credential presentation interaction mentioned above provides specialized example.

Delegation. In the offline world one entity can grant access to some resource to another entity. For example, I could give my car keys to a friend so they could borrow my car. There is no standard, or secure way to do this online. This is especially problematic in healthcare scenarios where a caregiver needs to gain access to electronic health-related data about another person.

Provider-defined Privacy Policies. Users are given the option to review the privacy policies put forth by the provider, policies which are designed to protect the provider's interests while staying within the limits defined by the relevant privacy regulations. The burden of making sense of these policies is shifted to the user (i.e. the potential victim) because the user doesn't have the time to read 100+ policies for the providers they typically use, nor do they have the computational power on their side to aid them in this assessment.

Privacy policy expression. With a few exceptions, (e.g. the Global Privacy Control[16]), users lack the technical means (i.e. computational power) to express their privacy terms to providers.

User rights. In a growing number of jurisdictions, starting with Europe's GDPR and expanding to other regions, the user's data rights, (e.g. the right to access, correct and delete their data), are explicitly stated. In principle these laws respect these rights, however in practice the time and effort required to exercise these rights is so exorbitant, that in practice they don't exist. The user has to send written requests to get their data, request that it by updated or deleted, etc. User-side agents are required for user's to regain in practice the rights they have in principle.

Inferences. "Eli Pariser need to give users control of their bubble" check this quote[5, p66](interests profile)

Feudalism. [You (as a peasant) work the fields, the landlord owns them and amass fortunes; mentions of Jared's and others AI injustice about who owns the work. Jared Lanier, etc.]

[16] globalprivacycontrol.org

1.3 Third-party surveillance

Whereas the user is at least aware when they sign up on a first-party app that their interactions are known to the provider of that app, there are hundreds of third-parties of which the user is unaware that track and assemble databases about them. Databases of user data in the hands of hundreds of unknown third-parties creates privacy risks and vulnerabilities. Users have little transparency into what's being gathered, where it's being shared and how it's being used. It is worth noting that that much of this third-party tracking is enabled in collaboration with first-parties (e.g. first-parties placing third-party tracking cookies on the user's browser).

Surveillance-based targeted advertising. Targeted advertising in general[17] involves four main processing steps: (1) the collection of observations about the user by a first- or third-party, (2) synthesis of an "ad profile" from these observations, (3) matching this ad profile against available "target audiences"(i.e. characteristics of whom the advertiser wishes to reach, advertising budget, etc.) from advertisers through a bidding process, and (4) displaying the winning ad. Surveillance-based targeted advertising specifically is when step (1) above is achieved by third-parties who track the user as they move from apps to app and site to site across the internet using third-party cookies and similar tracking mechanisms.

Third-party ad tech vendors perform the tracking and synthesis of a user's "ad profile." Users have no say in their own ad profiles–never seeing them and not the ability to correct them.

Data brokers. Data brokers who buy and sell personal data to other brokers, to advertisers, adtech firms and first-party publishers provide liquidity (along with a host of privacy threats) in the personal data shadow marketplace because user's lack the computational power to provide data about themselves.

1.4 Lack of Convenience

The prevailing architecture of the internet involves each provider managing their own information "silo" of information about the user (i.e. their account). This approach and the lack of computational power on the user's side creates inconvenience for them that is described below.

Repetition. When using apps, users ")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 user has the hassle of repeated data entry, the app has increased friction (a worsened user experience).

[17] Also known as behavioral advertising or more recently, interest-based advertising

Password management. The average user uses 100 websites and 25 apps daily. Managing and periodically updating strong, unique passwords at each is impractical without an automated password manager (computational power), yet it has been estimated that less than 5 percent of internet users use a password manager.

Account Management. The user has the inconvenient burden of maintaining the timeliness and consistency of their account information at over one hundred apps. For example, updating contact or credit card information at each is tedious, time-consuming and encourages the user to spend more time at sites that already have their information. The relative convenience of shopping on Amazon vs. another e-commerce site. It is partially caused by the user's lack of computational power to manage these relationships–the processes are not automated and tedious.

2 Design Considerations

Our vision is to develop an identity agent to address the problems outlined in the previous section. This agent can represent the user and promote their interests online. In this section, we discuss a number of design considerations for this agent illustrating the each with use cases.

2.1 User-centric vs. provider-centric

Many of the challenges described thus far have their origin in an architecture that is *provider-centric* rather than *user-centric*, or *human-centered*. In the provider-centric model each provider sees a single narrow slice of the user through the lens of their direct interactions with them. Each works in isolation to optimize the user's experience on their app or at their site. To the user the situation is reversed. They sit at the center of many dozens of connections radiating out from them to apps/sites. The user has the burden for entering, and updating information at each provider each of which maintains a separate copy.

The best technology so far to address this user burden is a browser form-fillers which greatly reduces the number of keystrokes required to fill in web forms. Browsers are *user-agents* that perform this operation on the user's behalf, and sit on the user's side of the power equation. Another closely related user-agent technology is that of password managers. In both cases these agents maintain small databases on the user's side.

2.2 Edge-centered vs. cloud-centered

Given that we need a per-user, user-centric decentralized architecture, where should this datastore live? We look at two options edge-centered and cloud-centered. Edge-centered means that the primary location for a user's personal datastore is on their own phones, and laptops and perhaps home servers. Cloud-centered means that the user's personal datastore

is primarily held in the cloud (e.g. on a SOLID[18] pod). We say *primary* because there are usually use-cases that involved replicating/syncing some of the data to the "other" location.

Security. Although some may disagree, it is our contention that having a personal datastore on a personal device is more secure than in the cloud. Even if each platform alternative where equivalently secure, a cloud-centered architecture aggregates millions of personal datastores at one service provider and thereby creates millions of times the economic incentive for hackers to invest in attacking it.

Equity. The hosting costs of a cloud-centered solution must be paid for by some entity whereas user's typically own their edge devices and they are thus on a marginal basis "free". By equity we mean here that we need a solution that can be afforded by all socio-economic classes, and a solution that requires monthly hosting fees can thereby be ruled out.

Backup. One serious drawback of what's called a *non-custodial* edge-centered architecture (when compared to cloud-centered) is the need for the user's data to be backed up. This is not a problem if the user backs up their devices (e.g. to a cloud backup service), but many can't be relied on to be disciplined about this. [stats on phone vs. laptop backup]. [we assume the user has N¿1 device and that the agent is installed on N¿1 and that the agents replicate/sync]

2.3 Replication

If we assume an edge-centered design, we must solve the roaming problem. That is, we must support use cases where the user has more than one device and needs to be able to pick any one of them and have their person datastore be consistent across these devices (at least eventually). This requires that the agents of a given user implement replication and syncing. [discuss the (unfortunate and costly) need for relays to achieve P2P sync in some use cases].

2.4 Loyalty

Much of the power asymmetry described in the first section is due to economic incentives for providers to do just enough in the user's interest to keep them as a user or customer, but not more. Personal data, after all, is considered by business to be an asset class; the more of it that is collected and monetized the better. To have an agent that works not partially but exclusively on behalf of the user, the agent provider must not have an economic incentive to provide anything less but complete loyalty to the user's interests. Although there are other potential solutions (e.g. data cooperatives and data unions) we think the simplest approach is that the agent developer be a nonprofit organization that

[18] solidproject.org

has no economic incentive to be anything but loyal to the user and to have no economic interest in their data. In fact there's no reason that that the agent developer to access or store the user's data.

2.5 Separation of app from db

[Talk about this from the point of view of the SOLID project or go-peer?? or ??. Also talk about how this is inevitable. Desktop OSes had it, then in 2007?? iPhone got it, now in 2023! webapps will get it. Talk about automatic data portability]

2.6 Delegation

[Talk about delegated medical records use case]. [Talk about the Gropper Principle]

2.7 Multi-contextual

Zuckerberg has said "Having two identities for yourself is an example of a lack of integrity" [3]. He could not be more wrong. Let's take a step back and look at selfness and whoness.

2.7.1 Selfness and Whoness

In his last public speech[19] Kim Cameron[20] 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.

The following diagram illustrates these concepts and introduces the notion of context:

[19] www.youtube.com/watch?v=9DExNTY3QAk

[20] en.wikipedia.org/wiki/Kim_Cameron_(computer_scientist)

2.8 Open source

In order to trust that the agent does what we claim, we need transparency which open source can provide. Further this is an ambitious project and we need to nurture the creation of a community of developers help build it.

2.9 Data Governance

Once data is shared from the agent to a first-party there are no technical means to constraint what the recipient does with it. No technical means can prevent them from selling it others, for example. Instead, legal means must be employed. Rather than wait for privacy regulations to get strong enough, we propose that first-parties sign a Human Information License to license the user's information under terms that are fair and balanced and respect the user's privacy rights. The contract can be signed by an entity that represents the user to make it effortless for the user.

2.10 User rights

[Talk about how even the best privacy legislation is impotent in practice to protect users because they don't require associated technical means to implement them]

2.11 Enforcement

Talk here about the need for an entity to audit and enforce compliance with the legal contract]

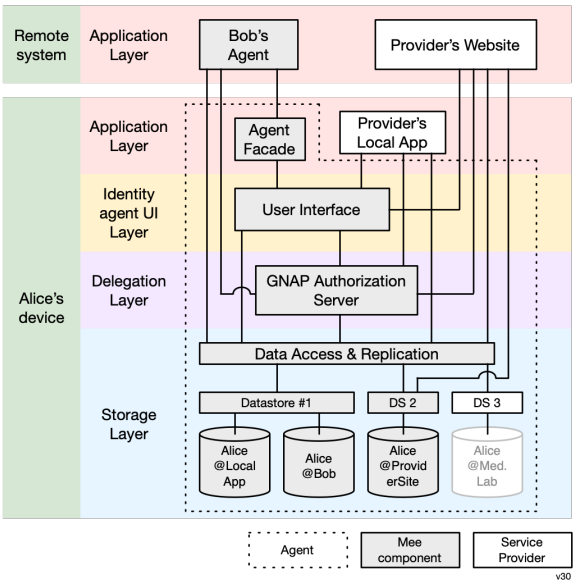
2.12 Privacy by design

3 Architecture

In this section we propose an architecture for identity agents.

3.1 Agent

The agent architecture follows a decentralized, layered architecture shown below. We illustrate this architecture by considering a user, Alice, with her own device (e.g. a smart phone) as well as three other parties: a provider's website, a provider's local app, and another user Bob's agent:



3.1.1 Application layer

Compatible applications in the application layer may be service provider websites ("Provider's Website" above), other user's agents running on their own devices ("Bob's Agent"), or a service provider's local app running on Alice's device ("Provider's Local App"). Alice's agent appears to other users' agents as an app. The component supporting this on Alice's side is called the Agent Facade.

3.1.2 Identity agent UI layer

Alice's identity agent is deployed as an app on Alice's device. The top of layer of the agent is the UI layer that provides Alice with data management features to connect with apps/sites and manage her data. This UI allows her to inspect and in some cases edit each of the partial representations of her in each connection's context(s).

3.1.3 Delegation layer

The delegation layer handles requests for access to data from Alice's agent, local apps, remote apps, and other users' agents using GNAP[21]. In response to these requests, Alice's authorization server grants or revokes access to data in the context data storage layer.

3.1.4 Storage layer

The data access and replication component provides data access (as controlled by authorization server above it) to the data in each of Alice's contexts. It manages the replication of changes to the data in one of Alice's contexts both (i) between the corresponding app and Alice's agent as well as (ii) among Alice's edge devices (phone, tablet, laptop, etc.).

This layer holds a set of contextualized representations of Alice as defined and created by apps/sites. The diagram above shows three local context data containers on Alice's device and one, the Med Lab app's context data container, which is not replicated on Alice's local device (perhaps because its data set is too large for Alice's device).

3.2 Private data sharing

Agents need to be able to bidirectionally share personal information with other parties in order to respect the user's data rights to access, correction (editing), and deletion of the information held by this party. The other party must respect these rights in order to become certified under the terms of the Mee Human Information License. To do so they must support the private data sharing communication protocols implemented by the agent. For authentication purposes the other party must implement the OpenID SIOPv2[22]. For bi-directional data sharing alternatives are currently being discussed.

References

[1] Klint Finley. Tim berners-lee, inventor of the web, plots a radical overhaul of his creation — wired. *Wired.com*, 4 2017.

[21] oauth.net/gnap/

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

[2] 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.

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

[4] 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.

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

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

[7] Carissa Véliz. *Privacy is Power: Why and how You Should Take Back Control of Your Data*. Random House, 2020.