EXPLORING CO-DESIGN CONSIDERATIONS FOR EMBEDDING PRIVACY IN HOLOCHAIN APPS: A VALUE SENSITIVE DESIGN PERSPECTIVE

Paul d'Aoust, Oliver Burmeister, Anisha Fernando, Anwaar Ulhaq, Kirsten Wahlstrom

Holochain (Canada), Charles Sturt University (Australia), University of South Australia (Australia), Charles Sturt University (Australia), University of South Australia (Australia)

oburmeister@csu.edu.au; paul.daoust@holo.host; Anisha.Fernando@unisa.edu.au; aulhaq@csu.edu.au; Kirsten.Wahlstrom@unisa.edu.au

ABSTRACT

Supporting privacy in contexts mediated by technology is a persistent challenge and this paper sets out to advance practice through value sensitive design and a data sharing platform called Holochain. Values shape the way people behave and interact and support the formation of identity. Value sensitive design supports designers, developers and others in attending to values when creating new technologies. One such value is privacy which in this paper is seen as intrinsic to (and shaping of) social contexts, and ubiquitous in varying degrees. The paper supports developers and others engaged in the co-design of Holochain applications by describing which features of Holochain can be used to support privacy, offering six technical features for doing so.

INTRODUCTION

When introducing their book on value sensitive design, Friedman and Hendry (2019) quoted Winograd and Flores, "in designing tools, we are designing ways of being" (Friedman & Hendry, 2019, p1). Value sensitive design supports developers, designers and others in attending to human values in the design of technologies (Friedman & Hendry, 2012). Values are important in the design of technology because values shape the way people interact and behave, as well as supporting us in shaping our identities (Alshehri, Kirkham, & Olivier, 2020). However, values do not exist in insolation and may be in tension, for example, if developers are working on short deadlines, the provision of accessibility may be in tension with expedience. The value attended to in this paper is privacy.

Privacy has been discussed for centuries. Aristotle differentiated the *oikos* from the *polis* and considered the head of a household responsible for representing the views of the private household in the public sphere (as described in Molitorisz, 2020). Later, John Stuart Mill wrote, "That there is, or ought to be, some space in human existence thus entrenched around, and sacred from authoritative intrusion, no one who professes the smallest regard to human freedom or dignity will call in question," (Mill, 1848, p938). Later still, Warren and Brandeis suggested that privacy was a right worthy of protection under the law (Warren & Brandeis, 1890). In 1929, Ragland perceived the challenges to privacy suggested by technologies and made the prescient suggestion that privacy may be of unusual importance in years to come, given the technological advances of his time (Ragland, 1929).

More recently, privacy has been variously described as the control of data in the pursuit of self-determination (Westin, 1967), restricting access to self and data (Moor, 1990), the capacity to trade (or otherwise) privacy and data as commodities (Posner, 1977), a social good unique to specific contexts (Burmeister, Islam, Dayhew, & Crichton, 2015; Nissenbaum, 2009), and as being shaped by the technologies forming the infosphere (Floridi, 2005). Today, privacy is understood to arise in diverse social contexts (Nissenbaum, 2009), taking diverse and pliant forms appropriate to those contexts (Wahlstrom, Fairweather, & Ashman, 2017) and is often mistakenly conflated with security and confidentiality (Mittelstadt, Fairweather, McBride, & Shaw, 2013; Wahlstrom & Fairweather, 2013). Regardless of whether technology mediates a social context, privacy is intrinsic to that context and pliant. Privacy is both shaped by social contexts and shaping of it (Wahlstrom et al., 2017), with some social contexts calling for no privacy while others call for much privacy. Thus, privacy is ubiquitous but in varying degrees, and frequently in tension with values such as seeking to serve shareholders by deriving market value from data assets.

In some cases, information technologies provide high levels of data privacy and in others, very little privacy is available. Regardless of how much or little privacy is afforded, pliancy in data privacy is rarely available. The right to be forgotten (Cellan-Jones, 2014) affords a type of pliancy in data privacy because under this right, someone may request that data be removed from search indexes and thus increase their data privacy. Wahlstrom, Ulhaq, and Burmeister (2020) considered the right to be forgotten in the context of an emerging peer to peer platform called *Holochain*. This paper extends that work by exploring considerations relevant to enabling privacy in Holochain applications (called *hApps*) through values sensitive co-design.

The next section provides an overview of data architectures followed by a description of Holochain, focussing on agency, privacy, and anonymity. This is followed by a discussion of how privacy can be supported with Holochain, which suggests six privacy design considerations for developers and eight community co-design considerations. This is followed by a section that discusses the six privacy design considerations in detail, prior to concluding the paper.

HOLOCHAIN

Applications that use a client/server architecture preserve data integrity by erecting a fortified wall around the data, consist of cybersecurity measures, data access policies, and validation and access control rules encoded in the application itself. However, the details of these protection measures are opaque, which may be unsatisfying for those seeking data privacy through agency over how the data describing them is used. Furthermore, this enclosure of data creates asymmetries in access to information, which in turn foster asymmetries in power (Janssen, Cobbe, & Singh, 2020).

With the introduction of two fundamental cryptographic tools: secure hashing algorithms and public-key encryption, distributed computing systems have become increasingly viable. These have offered solutions to key problems, such as verifiable, tamper-proof data for sharing state across distributed system nodes and data provenance validation using digital signature algorithms.

These systems can be divided into two types: data-centric and agent-centric. Data-centric systems aim to create a single shared data reality for all nodes (blockchains are examples of data-centric systems). Agent-centric distributed systems are concerned with allowing nodes to share constantly changing data (Zaman, Khandaker, Khan, Tariq, & Wong, 2021).

Blockchain is a transparent alternative to client/server applications, allowing equal access to both the data it stores and the details of the algorithms by which data integrity is preserved. However, this is possible only because all data is contained in a global public ledger that is replicated, inspected, and audited by all parties with a copy of the software that encodes the data integrity algorithms. Fine-grained access control to specific portions of the ledger is by definition impossible.

Holochain is an agent-centric distributed generalised computing system in which nodes can still participate in the system as a whole while not needing to maintain the same chain state as the other nodes (Harris-Braun, Luck, & Brock, 2018). A Holochain application (a *hApp*) "… consists of a network of agents maintaining a unique source chain of their transactions, paired with a shared space implemented as a validating, monotonic, sharded, distributed hash table (DHT) where every node enforces validation rules on that data in the DHT as well as providing provenance of data from the source chains where it originated" (Harris-Braun et al., 2018, p4). In Holochain's DHT and its proof by enforcement. all elements of the DHT can only be modified monotonically; that is, elements can only be added to DHT and not removed.

Holochain and agency

Holochain starts with a fundamentally different perspective from either blockchain or client/server architectures: an acknowledgement that all information has its origin in the subjective experience of the agent (usually a person) producing it, and the argument that data separated from its provenance has lost a critical part of its meaning (Harris-Braun, 2021). Holochain is designed around this perspective.

In putting agency rather than data at the centre of Holochain, there was no intention to explicitly enhance individual privacy or freedom, however these features are natural side-effects of this orientation (Harris-Braun, 2021).

Each participant in the system exercises their agency via a computer device under their control, recording their actions as entries on a personal digital journal. These entries are cryptographically signed by a private key, which provides the aforementioned provenance context. When the application requires them to share these entries publicly, they do so by broadcasting to a subset of their peers who use the same application. Those peers then take responsibility for validating, storing, and serving the entries to others.

Holochain and privacy

There is a dynamic tension between individual and collective agency, which application developers and other stakeholders must consider in their designs. This becomes especially apparent with respect to privacy. When an individual keeps their information secret, it is naturally as private as possible. However, a networked application is useful precisely because it allows information to be shared. This requires the individual to divulge some of their information in order to derive utility from the application.

When someone discloses their information to others, whether peers in a peer-to-peer digital network, to a centralised server, or to other people, it is witnessed and becomes in one sense part of a collective memory. It may never be completely forgotten; even if it is diligently removed from all digital databases, backups, and caches, it may still live on in the memories of those who saw it.

The medium in which a piece of information is preserved imparts an amount of 'friction' (Floridi, 2005) or 'greasing' (Moor, 1990) to the movement of that information, regardless of whether the medium is a technological context. When someone is asked to forget information shared with them, the intention may not be to request they remove it from their memory, but to refrain from circulating it in low-friction media; for example, to refrain from speaking it to others. This results in enhanced privacy because the movement of the information is impeded.

There may also be cases where 'my information' cannot be distinguished from 'our information' without damaging the integrity of dependent information and thus causing material harm. An example is a ‘mutual credit’ or ‘barter exchange’ network, in which each member keeps their own ledger of debits and credits. Often, ledgers are visible to all other members, in order to facilitate mutual auditing and credit checking.

Consider a scenario in which Alice transfers some of her credit to Bob in exchange for goods, which gives Bob a positive balance that he can then spend with others. In a Holochain-based implementation, each transaction involves Alice and Bob mutually creating and countersigning a transaction record, which is then written to their respective ledgers. The transaction record, as a product of their interaction, belongs to both of them. If Alice were to leave the system and ask Bob to ‘forget’ her transaction with him, Bob’s ledger would no longer appear to have sufficient positive balance to spend.

In light of these privacy issues in a peer-to-peer network, it can be suggested that the most optimal privacy-preserving designs will be highly tailored to the social context of the group for whom the application is developed. This coheres with the intent of a Holochain application: if the ostensible aim of the executable code is to embody the norms of a group of users, then strong feedback loops should exist between those users and the developers writing the code, ideally using a participatory design process such as the Value Sensitive Action-Reflection Model described below.

Holochain and anonymity

On the other hand, Holochain does not easily support privacy through anonymity, as it was designed to support accountability among individuals who consent to participate in a context. An agent ID (the public key of an asymmetric key pair) is a long-lived identifier. Although the Holochain protocol does not prevent users from creating an arbitrary number of anonymous IDs, an application that permits this may be more vulnerable to an attacker running multiple nodes in an attempt to overwhelm the network’s ability to assure data integrity; a Sybil attack (Douceur, 2002).

A Holochain application consists of a set of rules that define valid actions an agent can take. This is not unique; almost all software that deals with user input incorporates validation rules. What makes Holochain and other peer-to-peer protocols unique, especially compared to client/server applications, is that every participant has a copy of these rules on their own device. This allows them to ensure that their own actions are valid and to check the validity of others’ actions. What emerges from these two properties is that the individual agents using the application form a cohesive group, defined by their mutual consent to be bound by the application’s validation rules. These rules become a digital, executable encoding of the group’s norms (Lessig, 2000). For this reason, the Holochain protocol is described as *social DNA* and the core executable code of a Holochain application is called a DNA bundle.

SUPPORTING PRIVACY WITH HOLOCHAIN

hApps have the potential to support privacy because Holochain provides scope for the appropriate consideration of social contexts. As noted in the introduction, social contexts give rise to privacy, regardless of whether a context is mediated by technology. Key privacy by design considerations in hApps can be embedded by creating and configuring contexts that uphold the privacy expectations of social contexts, such as the right to be forgotten. From a developer’s perspective, there are six design considerations, namely:

Entry visibility Defining data types as private, public, or public but encrypted in data schemes.

Membership membranes Participants' access to a hApp's network space can be specified in code given that all public data is visible in this space.

Partitioned data Use of two sharding methods: each party maintains their own journal, and each separate entry and header in a journal can be viewed, validated and stored by a small sample of other participants.

Capability-based security Access to private entries on a journal can be approved via use of capability tokens that may be non-transferable.

API membrane Only public data is verified or approved by the network, with the definition of valid data being determined by the developer and the participants' contexts.

Withdraw and purge Two new techniques to remove data are theorised with the implementation details pending: withdraw (redact data formerly published) and purge (offer a participant to mark another's data for deletion).

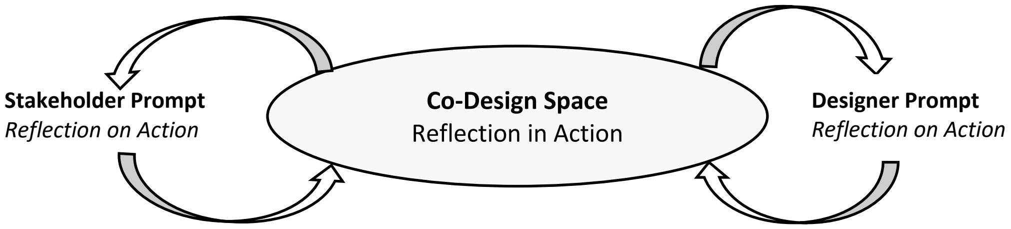
However, while developer requirements are important, the requirement to consider design needs from the community of people running or accessing hApps supports the balancing of data integrity with privacy. This ethical tension can be addressed by applying the value sensitive action-reflection model, a value sensitive design tool (Friedman & Hendry, 2019), see Figure 1. We suggest using the value sensitive action-reflection model recommended by Yoo et al. (2013) for the co-design of hApps in order to address the privacy needs arising in social contexts. The development of designer and stakeholder prompts is noted as a future research opportunity.

For example, the design needs of the people running or potentially accessing hApps should be considered. One approach to raise these design considerations draws on the core ICT values of privacy suggested by Huldtgren (2014): security, ownership, universal usability, autonomy, trust, accountability and human welfare. Hence, when developing a co-design space for hApps, these community-based design considerations could be explored:

Privacy This value is of core focus to this social context, and privacy-by-design criteria are embedded as listed in the developer considerations above. hApps have the potential to appropriately apply contextual privacy specific to social contexts. To understand the nuances of these social contexts when designing and to increase uptake across practical applications, participants may need data ethics literacy skills. Efforts to grow the awareness and implementation of privacy-related data knowledge and skills are needed to maintain the integrity of hApps.

Security What are potential security risks that need to be considered from a hApp community practitioner's perspective? For example, if social engineered cyber-attacks have the potential to input unauthorised data into the holochain, some privacy-by-design safeguards need to be considered when codesigning in practice.

Figure 1. Evolving the Co-Design Space for Holochain, adapted from Yoo, Huldtgren, Woelfer, Hendry, and Friedman (2013).



Ownership As a disruptive approach to managing and storing data, how is this decentralised form of data ownership perceived by potential community users? For example, community-based hApps may need to be designed with common public-access protocols in place both from technical and human perspectives.

Universal Usability Can anyone use a hApp or are specific types of knowledge and skills required? How can these competencies be inclusively acquired? For example, if the hApp is designed for a context, considerations for accessibility and inclusiveness may need to be incorporated in subsequent iterations of the hApp.

Autonomy How can hApp users practice autonomy over managing data choices? For example, users may need to think of and use the data embedded as having collective privacy (i.e. “our data”) as opposed to individual privacy (i.e. “my data”).

Trust What ensures the trustworthiness of hApp and the integrity of data managed in these transactions? For example, end-users, designers and decision-makers will need to adapt to new practices and measures of trustworthiness and integrity pertaining to the data held in these hApps.

Accountability and Responsibility Which stakeholder(s) are accountable and responsible for the integrity of these hApps? What are mechanisms to practice these values? For example, organisations developing hApps may need to develop capability by having digital or data ethics designers incorporated in their project development teams to embed privacy-by-design principles.

Human Welfare Are there any potential harms propagated or introduced by managing data in hApps? For example, the value sensitive action reflection model is useful to apply in this context to reflect on potential instances where the contextual nature of privacy is upheld, disrupted or strained such as via a value tension. The consideration of these central ICT values is not an exhaustive list but is recommended as a useful place to start in unpacking the ethical tensions in emerging technologies such as hApps. The potential of these emerging technologies to appropriately address the need for embedding social contexts for privacy and further the contextual relational nature of privacy should be explored.

SIX PRIVACY DESIGN CONSIDERATIONS

To support the value sensitive design of hApps, co-designers should have a strong grasp of the basic building blocks of Holochain and how they affect privacy, as well as the tensions between privacy and data integrity mentioned above. This knowledge ought to serve as foundations for useful design prompts to aid in the participatory design process. This section details the six design considerations that affect privacy that were identified above.

Entry visibility

An individual's journal consists of entries linked together cryptographically by headers. For each entry (not including system entries), a data type or schema can be defined, along with the type's *visibility*. The visibility attribute defines whether entries of this type are shared with a random subset of peers who are using the same application. These peers are called *authorities,* by virtue of having been selected to validate, store, and make the data available to any peer who has a reference to it. This mechanism builds a distributed hash table (DHT) of shared data among the users of the application. The two visibility options for an entry type are:

> *Private*, in which only the header is shared, while the entry content is kept locally on the individual's device; or
>
> *Public*, in which both the entry header and content are shared.

The distinction between these two can be compared to a card game in which some cards are dealt face-up and some are dealt face-down for one player to put in their hand. All players can see every card that has been dealt; not all players can see the contents of all cards (Brock, 2009).

The availability of this option stands in contrast to most public blockchains, which typically require all data to be made public for validation by all peers. While metadata can be written to blockchain transactions to refer to the existence of off-chain data, similar to private entries in Holochain, this is an ad-hoc solution that requires integration with another technology in order to store said data.

There are two compromises that must be made in deciding between private and public visibility; this is noted as possible design guideline. First, private data suffers from reduced availability, as it requires the author to be online if any peer needs to request it. Second, it cannot be publicly validated, so it cannot contribute to the network's aggregation of metadata that indicates what and who can be trusted.

As previously mentioned, as a third option public data can also be encrypted before sharing in order to 'hide it in plain sight'. Available to blockchains as well, this option allows data to be accessed by others who know how to decrypt it, even when the author is offline. Traditional symmetric-key encryption can be used, although this once again prevents authorities from being able to validate it.

Novel encryption schemes, such as zero-knowledge proofs (eg ZK-SNARKS, pioneered with the ZCash blockchain), allow data to remain secret while still being subject to validation; this is also an option for Holochain applications. However, such encryption schemes may incur high computational overhead and not all validation problems can be modelled with zero-knowledge proofs.

Membership membranes

Holochain anticipates the need to define appropriate privacy norms for different contexts. As each DNA bundle and its network of users represent a context, it is possible to define a ‘membrane’ that determines who does and does not belong to the context. This is implemented via a validation function on an entry near the beginning of a journal, the ‘joining proof’. This proof can contain credentials such as an invite code, an attestation signature from an existing member, or a proof of identity or membership in the corresponding human network.

Context membership can be further managed by ejecting an individual from the context. This may be required as a response to the intentional production of invalid data, which would indicate an individual has tampered with their copy of the software. It can also be used in non-adversarial situations, such as an employee leaving a company and having their access to company data revoked.

Partitioned data

Data in Holochain is partitioned in three ways. Firstly, as noted above, the individual’s journal gives them the power to create their own data and permits them to keep some of that data unpublished.

Secondly, each application has its own network with its own store of public data (DHT). Here, ‘public’ indicates accessibility to other authenticated context members within the application’s membrane, not accessibility to the world. Networks are isolated from each other and distinguished by the cryptographic hash of their DNA bundles’ executable code. This can also be exploited to create multiple separate contexts operating with the same executable code, by changing an insignificant detail such as the application’s title or ID that causes the hash to change.

Thirdly, the authorities selected to validate, store, and serve each piece of public data are only a subset of the entire context, and are selected randomly. Any and likely all agents will be selected at various times to be authorities for at least some pieces of public data. This distribution, called sharding, scatters data throughout the network. This reduces the amount of data that any agent is able to easily analyse. Additionally, a piece of data is referenced via an opaque ID (the cryptographic hash of the content) and cannot be retrieved by an agent unless they know this ID.

These are weak privacy measures, however, as any agent can enlist themselves as an authority for an arbitrarily large portion of the public data set. It does, however, place a greater computation and storage burden on such a peer, which could potentially serve as a deterrent to anyone who wishes to capture and analyse the entire data set.

Capability-based security

An individual can generate *capability grants* that give selective permission for peers to call specified public functions in their running DNA instance. In a sense, the individual is delegating some of their agency to others, as these functions have all the privileges the owner of the application instance enjoys. The developer can use this feature to, for example, create ‘getter’ (accessor) functions that return private journal entries.

A capability can be granted with one of three levels of access restriction:

- An *unrestricted* grant allows anyone to call the function for which the capability is being granted and is often applied to functions that allow peers to negotiate the granting of capabilities with tighter access restriction.

- A *transferable* grant is coupled with a secret token, and only permits peers who possess the token to call the function.
- An *assigned* grant is coupled with both a secret token and a list of peer IDs, and only permits access for peers who possess the token and whose ID appears in the list.

These grants can be created and revoked at any time, according to the functionality that the developer has designed into the application. This allows much finer-grained access control than the simple public-versus-private visibility described in point 1, although data protected and accessed in this manner is private and thus subject to the compromises described in that section.

API membrane

An application's DNA bundle consists primarily of functions that abstract over the low-level Holochain functionality and present a coherent view of the underlying data. In this sense, such functions can be considered a data access layer or API. Their chief purpose is to allow the owner of a running application instance to exercise their agency via a user interface (although, as point 4 mentions, capability tokens can be used to delegate that agency to others).

It must be noted that, when data is published to the DHT, it cannot properly be deleted or modified. This is due to the fact that the DHT is a monotonically increasing set that merely aggregates data points; this makes it easier to replicate data (Hellerstein & Alvaro, 2019) and maintains the historical record necessary to ensure distributed data integrity. However, data can be marked as *deleted* or *updated* and this appears as a piece of metadata attached to the entry that marks it as obsolete and, in the case of *updated*, points to a replacement entry.

When accessing public DHT data, the DNA bundle's functions can manipulate the result set before returning it to the UI. Specifically, they could be written to honour the deleted or updated status of entries and filter them from the return value. While this does not guarantee that the data is inaccessible, it does require a considerable amount of technical skill to bypass the Holochain runtime and access the raw data. In light of the view that privacy is enhanced and eroded in part by friction/greasing, it may not be absolutely necessary to completely eliminate the data in question; the aforementioned could potentially satisfy, for example, the right to be forgotten in an acceptable fashion for some applications.

Withdraw and purge

The Holochain protocol implements two new operations for the shared database: *withdraw*, which allows an author to ask validators to remove data they mistakenly published; and *purge*, which allows anyone to mark anyone else's data for removal. These are intended to be 'true' deletion operations for public DHT data, causing all compliant peers to honour the requests by erasing their copies of the data from their devices. While these operations are often used for the removal of errors or illegal content, we anticipate that they could also be used to exercise contextual privacy norms in a way that aligns more closely with a user's natural expectations. It should be noted, however, that in a peer-to-peer system these operations merely constitute a polite request and cannot be algorithmically forced upon non-compliant peers.

We also note that such capacities may be incompatible with the needs of a given context. As seen previously in the example of a mutual credit network, data that becomes part of the public record, i.e., ‘our information’, can become highly interrelated. In these cases, removing one entry might cause entries that depend on it to become invalid. Additionally, even for data that exists independently of other data, the context’s rules may preclude that possibility, such as for a federal voting application that requires immutable records in order to confirm the results of an election.

Finally, a meta-consideration involves the upgrading of context rules. Effective privacy adapts to changing social contexts, so any privacy rules embedded in a DNA bundle also need to adapt if they are to serve evolving or emerging privacy norms. As a peer-to-peer application only exists when there are at least two individuals actively running instances of it, those individuals can agree to upgrade to a new application with a new set of rules, split into multiple groups, or elect to stay with the old rules if they prefer them. This can be assisted through upgrade routines in a DNA bundle, which may require that sufficiently flexible group governance processes for upgrades be designed into the first iteration of the application. This may offer a more flexible option than public blockchain platforms which offer an unchangeable set of base rules and do not allow application-specific rules (smart contracts) to be modified, although we also note that blockchain developers are beginning to establish design patterns that allow smart contracts to be retired or superseded by new versions.

CONCLUSION

Values are integral to the way people interact and behave. Value sensitive design supports developers and others in attending to values as they undertake the co-design of technologies. With respect to hApps, it is important to focus attention on privacy, security, ownership, universal usability, autonomy, trust, accountability and responsibility, and human welfare.

The support of privacy in contexts mediated by technology has been a persistent problem, with regulatory approaches failing to support contextual nuances and evolving privacy norms, and technologies failing to offer sufficient diversity and flexibility in privacy options. Unlike Blockchain which does not support privacy effectively, Holochain is a data sharing technology offering six features that may be leveraged by developers to create applications that support nuanced and contextual privacy norms while sharing data in non-repudiable ways. A future research opportunity is the investigation of the extent to which these features can be successfully leveraged in value sensitive design co-design projects.

KEYWORDS: value-sensitive design (VSD), privacy, Holochain.

REFERENCES

Alshehri, T., Kirkham, R., & Olivier, P. (2020). *Scenario Co-Creation Cards: A Culturally Sensitive Tool for Eliciting Values.* Paper presented at the Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems.

Brock, A. (2009). Differences Between "Open Source" and "Open Currency". Retrieved from https://www.artbrock.com/2009/05/12/differences-between-open-source-and-open-currency#digging-deeper-into-open-data

Burmeister, O. K., Islam, M. Z., Dayhew, M., & Crichton, M. (2015). Enhancing client welfare through better communication of private mental health data between rural service providers. *Australasian Journal of Information Systems, 19*. https://doi.org/10.3127/ajis.v19i0.1206

Cellan-Jones, R. (2014). EU court backs 'right to be forgotten'in Google case. *BBC News (14 May 2014) online: BBC News Europe* http://www.bbc.com/news/world-europe-27388289

Douceur, J. R. (2002). *The sybil attack.* Paper presented at the International workshop on peer-to-peer systems.

Floridi, L. (2005). The Ontological Interpretation of Informational Privacy. *Ethics and Information Technology, 7*(4), 185-200. https://doi.org/10.1007/s10676-006-0001-7

Friedman, B., & Hendry, D. (2012). *The envisioning cards: a toolkit for catalyzing humanistic and technical imaginations.* Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.

Friedman, B., & Hendry, D. (2019). *Value sensitive design : shaping technology with moral imagination*. Cambridge: MIT Press.

Harris-Braun, E. (2021). Decentralized Next-level Collaboration Apps with Syn. Retrieved from https://blog.holochain.org/decentralized-next-level-collaboration-apps-with-syn/#back-story-agent-centric-data-enables-collaboration

Harris-Braun, E., Luck, N., & Brock, A. (2018). Holochain: scalable agent-centric distributed computing. Retrieved from https://github.com/holochain/holochain-proto/blob/whitepaper/holochain.pdf

Hellerstein, J. M., & Alvaro, P. (2019). Keeping CALM: when distributed consistency is easy. *arXiv preprint arXiv:1901.01930*.

Huldtgren, A. (2014). Design for Values in ICT. In M. J. Van Den Hoven, P. E. Vermaas, & I. van de Poel (Eds.), *Handbook on ethics, values and technological design: sources, theory, values and application domains* (pp. 1-24). Netherlands: Springer.

Janssen, H., Cobbe, J., & Singh, J. (2020). Personal information management systems: A user-centric privacy utopia? *Internet Policy Review, 9*(4), 1-25.

Lessig, L. (2000). Code is law. *Harvard magazine, 1*(2000).

Mill, J. S. (1848). Principles of political economy with some of their applications. *Social Philosophy*.

Mittelstadt, B., Fairweather, B., McBride, N., & Shaw, M. (2013). *Privacy, risk and personal health monitoring.* Paper presented at the ETHICOMP, Kolding, Denmark.

Molitorisz, S. (2020). *Net Privacy: How we can be free in an age of surveillance*: NewSouth.

Moor, J. (1990). The ethics of privacy protection. *Library Trends, 39*(1), 69-82.

Nissenbaum, H. (2009). *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford, US: Stanford Law Books.

Posner, R. (1977). The right of privacy. *Ga. L. Rev., 12*, 393.

Ragland, G. (1929). The Right of Privacy. *Kentucky Law Journal, 17*, 85-122. doi:citeulike-article-id:13666094

Wahlstrom, K., & Fairweather, N. B. (2013). *Privacy, the Theory of Communicative Action and Technology*. Paper presented at the ETHICOMP, Kolding, Denmark.

Wahlstrom, K., Fairweather, N. B., & Ashman, H. (2017). *Brain-Computer Interfaces and Privacy: Method and interim findings*. Paper presented at the ETHICOMP, Turin, Italy.

Wahlstrom, K., Ulhaq, A., & Burmeister, O. (2020). Privacy by design. *Australasian Journal of Information Systems, 24*.

Warren, S., & Brandeis, L. (1890). The Right to Privacy. *Harvard Law Review, 4*(5), 193-220. doi:citeulike-article-id:6597501

Westin, A. (1967). *Privacy and Freedom*: Atheneum.

Yoo, D., Huldtgren, A., Woelfer, J. P., Hendry, D. G., & Friedman, B. (2013). *A value sensitive action-reflection model: evolving a co-design space with stakeholder and designer prompts*. Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.

Zaman, S., Khandaker, M. R., Khan, R. T., Tariq, F., & Wong, K.-K. (2021). Thinking Out of the Blocks: Holochain for Distributed Security in IoT Healthcare. *arXiv preprint arXiv:2103.01322*.