Libertas: Privacy-Preserving Collaborative Computation for Decentralised Personal Data Stores

RUI ZHAO, University of Oxford, United Kingdom
NAMAN GOEL, University of Oxford, United Kingdom
NITIN AGRAWAL[*], University of Oxford, United Kingdom
JUN ZHAO, University of Oxford, United Kingdom
JAKE STEIN, University of Oxford, United Kingdom
WAEL ALBAYAYDH, University of Oxford, United Kingdom
RUBEN VERBORGH, Ghent University, Belgium
REUBEN BINNS, University of Oxford, United Kingdom
TIM BERNERS-LEE, University of Oxford, United Kingdom
NIGEL SHADBOLT, University of Oxford, United Kingdom

Data and data processing have become an indispensable aspect for our society. Insights drawn from collective data make invaluable contribution to scientific, societal and communal research and business. But there are increasing worries about privacy issues and data misuse. This has prompted the emergence of decentralised personal data stores (PDS) like Solid that provide individuals more control over their personal data. However, existing PDS frameworks face challenges in ensuring data privacy when performing collective computations that combine data from multiple users. While Secure Multi-Party Computation (MPC) offers input secrecy protection during collective computation without relying on any single party, issues emerge when directly applying MPC in the context of PDS, particularly due to key factors like autonomy and decentralisation. In this work, we discuss the essence of this issue, identify a potential solution, and introduce a modular system architecture, Libertas, to integrate MPC with PDS like Solid, without requiring protocol-level changes. We introduce a paradigm shift from an 'omniscient' view to individual-based, user-centric view of trust and security, and discuss the threat model of Libertas. Two realistic use cases for collaborative data processing are used for evaluation, both for technical feasibility and empirical benchmark, highlighting its effectiveness in empowering gig workers and generating differentially private synthetic data. The results of our experiments underscore Libertas' linear scalability and provide valuable insights into compute optimisations, thereby advancing the state-of-the-art in privacy-preserving data processing practices. By offering practical solutions for maintaining both individual autonomy and privacy in collaborative data processing environments, Libertas contributes significantly to the ongoing discourse on privacy protection in data-driven decision-making contexts.

[*] This author contributed to this work while being affliated with the University of Oxford

Authors' Contact Information: Rui Zhao, University of Oxford, United Kingdom, rui.zhao@cs.ox.ac.uk; Naman Goel, University of Oxford, United Kingdom, naman.goel@cs.ox.ac.uk; Nitin Agrawal, University of Oxford, United Kingdom, nitin.cic@gmail.com; Jun Zhao, University of Oxford, United Kingdom, jun.zhao@cs.ox.ac.uk; Jake Stein, University of Oxford, United Kingdom, jake.stein@cs.ox.ac.uk; Wael Albayaydh, University of Oxford, United Kingdom, wael.albayaydh@cs.ox.ac.uk; Ruben Verborgh, Ghent University, Belgium, Ruben.Verborgh@UGent.be; Reuben Binns, University of Oxford, United Kingdom, reuben.binns@cs.ox.ac.uk; Tim Berners-Lee, University of Oxford, United Kingdom, tim.berners-lee@cs.ox.ac.uk; Nigel Shadbolt, University of Oxford, United Kingdom, nigel.shadbolt@cs.ox.ac.uk.

CCS Concepts: • **Computer systems organization → Distributed architectures**; • **Security and privacy → Security services**; • **Human-centered computing** → *Collaborative and social computing*.

Additional Key Words and Phrases: Solid, Personal Data Store, Decentralised Web, Collaborative Computation, Privacy, Multi-Party Computation

ACM Reference Format:
Rui Zhao, Naman Goel, Nitin Agrawal, Jun Zhao, Jake Stein, Wael Albayaydh, Ruben Verborgh, Reuben Binns, Tim Berners-Lee, and Nigel Shadbolt. 2025. Libertas: Privacy-Preserving Collaborative Computation for Decentralised Personal Data Stores. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 29 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Data is a valuable resource for science and business, as well as for society and individuals. There is a growing imperative to leverage collective data for societal benefits. Examples include enhancing pandemic responses by aggregating health records [100], addressing climate change through energy consumption data exchange [10], and even ameliorating working conditions by pooling telemetry data among otherwise isolated gig workers [98, 107].

Previous CSCW research has extensively discussed the benefits of user-centric or user-led approaches to collaboration in extracting collective insights [23, 46, 64, 112], which provided evidence and suggested the necessity of helping users regain control over their data and achieving better data autonomy. However, most often, they still make use of a trusted central party (the system) for data storage, data management and computation. This can lead to issues and concerns similar to existing centralised platforms, such as barrier of account registration [48, 50, 69], lack of data portability [49, 95], and trust and abuse concerns [7]. A truly user-centric design should eliminate such dependencies and their associated concerns, fostering better user adoption and greater flexibility in engaging with or withdrawing from both new and existing collective computational tasks.

Two recent developments that have gained increasing attention could form potential foundations for addressing these issues: decentralised Personal Data Stores and privacy-preserving computation. Personal Data Stores (PDS) [32, 78, 92] are considered a promising decentralised approach to addressing the privacy and autonomy concerns faced by individuals due to centralised platforms [61, 113]. They promote a paradigm shift from *platform-centred* architectures to *user-centric* architectures, where users store their data in a trusted location (the PDS), and applications or services request access to such data from the PDS, rather than storing and managing data on their own servers. However, this model still presents privacy challenges when collaborative computation is required on data distributed across multiple PDSs (see Section 2.2). Thus, algorithmic mechanisms for privacy-preserving computation offer a promising choice to address these challenges with mathematical guarantees. In particular, Secure Multi-Party Computation (MPC) [67, 104] enables the secure evaluation of functions over multiple parties' data without revealing their inputs. This can be further complemented by Differential Privacy (DP) [39], which aims to mitigate concerns about inferring inputs from computational outputs. However, discussions in the relevant fields have so far mainly focused on potential benefits, such as data quality and performance. As we will see later (Section 2.3 and Section 3.1), challenges remain in a user-autonomous setting.

This paper seeks to answer a natural yet overlooked question: Is it feasible to perform collaborative computation in a decentralised PDS context while respecting ethical aspects of privacy? In particular, can we go beyond the data secrecy / confidentiality aspect of privacy, and also include users' varying preferences for control, autonomy and trust as dynamically customisable factors in a system?

Table 1. Identified crucial ethical and functional requirements for a general collective privacy-preserving computation system in decentralised contexts.

Requirement	Explanation
R1: Data Privacy	Keeps (input) data confidential for the computation, not revealing to third parties
R2: User Autonomy	Gives meaningful control to the data providers/owners over permitted usages of their data
R3: User-centric Trust	Respects heterogeneous trust preferences by different users, and modifies system behaviours accordingly
R4: Scalability	Scales well with the number of data providers
R5: Generality	Supports various types of computations

We identify several core ethical and functional requirements that such a solution should address, as summarised in Table 1. As we will discuss in Section 2.1, we incorporate additional ethical dimensions of privacy into technical systems: the traditional CS aspect of confidentiality is referred to as the (input) **data privacy (R1)** requirement, which regulates the computational mechanism to not unexpectedly leak data; *user's will and control* is referred to as **(user) autonomy (R2)**, which expresses additional constraints that the technical solution should satisfy when performing computations; we also highlight the need for **user-centric trust (R3)**, that the system should respect and not coerce different users' heterogeneous trust preferences. In addition, two functional requirements have also been identified: the system should possess good **scalability (R4)** with an increasing number of participants, to be deployable in the real world; likewise, the system should be generic (**generality** requirement **R5**) to support various (if not all) types of computation, rather than being specialised to a limited number of computational jobs. Of course, there are other aspects that a system can support, but we consider the said requirements as the key requirements for a generic system (see also Section 2.1 for further critical discussions about requirements). We demonstrate in this paper that our proposed system, Libertas, can address the five requirements, and also support additional aspects, such as *output privacy*, in addition to the *input privacy*, enabling further uses.

In the remainder of this paper, we identify the shortcomings of current practices (Section 2) and the challenges of combining MPC with PDS (Section 3.1). These challenges stem from the Autonomy (R2) and Minimum Trust (R3) requirements, and the involvement of numerous autonomous data providers not typically encountered in centralised settings. We introduce Libertas (Section 4), a pioneering modular architecture that addresses **how** to combine MPC with PDS to support collaborative privacy-preserving computation in decentralised contexts, without requiring changes to underlying protocols. We demonstrate our implementation based on Solid [92], and discuss its adaptability to other PDS systems. . Individual-based, user-centric trust is a unique property of Libertas, where trust preferences are specified by users individually. We discuss its implications in the threat model section (Section 4.3). Libertas utilises the delegated-decentralised MPC model, offering superior scalability (linearly with the number of data providers) compared to the direct-decentralised model, as validated through benchmarking (Section 5.1). We further evaluate Libertas with two realistic example use cases (Section 5.2): 1) gig workers performing collective computation on their earnings data; and 2) synthetic differentially private data generation for privacy-enhancing public data release (e.g. census data). These evaluations confirm the technical feasibility and scalability of our approach, showcasing its broad applicability and potential.

Contributions: This paper makes both theoretical and practical contributions, as summarised below:

(1) It highlights a distinct challenge and gap in employing Secure Multi-Party Computation (MPC) in decentralised user-autonomous contexts such as Personal Data Stores (PDS);

(2) It proposes a novel technical solution, Libertas[1] , to address the challenge, and highlights the necessity and implications of individual-based, user-centric security and trust, in contrast to the ‘omniscient’ view in existing literature;

(3) Through two case studies of collaborative computations – gig worker empowerment and synthetic data generation – it evaluates the proposed system as follows:
   - Theoretically, it demonstrates how Libertas can contribute to CSCW for achieving collective benefits while strengthening individual privacy and autonomy simultaneously. Our discussion in the gig worker case study also considers how real-life power dynamics can establish the required trust relationships in Libertas to foster adoption;
   - Practically, it validates the implementation, demonstrates scalability patterns, and identifies practical insights.

2 Background and Related Work

As our work aims to bridge a gap between socio-technical discussions and technical solutions regarding privacy, and because there are some differences in terminology used in the fields, this section first presents a general discussion of these differences, and then introduces the technologies relevant to our research.

2.1 Privacy and Trust

Different aspects of privacy. The concept of privacy has been widely researched, and interdisciplinary scholars have noted the difficulty of precisely defining privacy [33, 81, 83, 88, 96]. In socio-technical systems, (information) privacy practices often include informational self-determination, the autonomy to control access to the information about self, and also contextual norms [8, 33, 81, 96].

Computer science research often considers security and privacy simultaneously, focusing on a narrower aspect of (information) privacy: information (data) confidentiality. Examples include cryptography [35], multi-party computation [67], and differential privacy [39], where ‘privacy’ refers to data (or other types of information) being kept secret (accessible only to the “owner" or other prescribed parties as discussed in the respective technology). While not sufficient on their own, these privacy-enhancing technologies (PETs) can provide individuals with a complementary means of protecting their personal information, while also enabling them to use that information for individual and collective benefit.

In this work, we attempt to bridge the gap between socio-technical discussions of privacy (specifically, autonomy or self-control) and practices in computing (specifically, confidentiality of data used in computation), as summarised in Table 1. Note that we distinguish autonomy from the other privacy aspects conventionally discussed in computer science literature, using a different term to highlight this gap; this is not to imply that autonomy is an entirely separate concept from privacy in the broad sense. In later sub-sections, we provide details on contemporary work regarding personal data stores (PDS), a promising paradigm for enabling individuals to exercise autonomy over their personal data, and PETs (such as multi-party computation and differential privacy), technical solutions for ensuring data confidentiality.

Privacy and user behaviours. Aside from discussions of the concept of privacy, there is also a related debate about ‘privacy paradox’ [11, 13, 45, 60], where users’ stated privacy preferences may diverge from their actual behaviour

[1] Our prototype implementation is available at https://github.com/OxfordHCC/libertas.

and their willingness to utilise privacy controls, especially in social media and e-commerce contexts. This is also connected to the research on privacy calculus and related topics [4–6, 34, 36, 45, 74], which considers privacy-related decision-making as a complex mechanism including other factors such as perceived benefits, costs, risks, etc. In that regard, our work does not directly answer the questions about users' privacy protection behaviour. Instead, it focuses on discussing and providing novel technical solutions that can potentially **influence** the benefits, costs, and convenience of use, and thus users' perceptions and adoption of user-empowering and/or privacy-enhancing technologies.

Trust. Similar to privacy, trust is also a multifaceted concept with diverse definitions [28, 71, 99], and multiple factors can affect users' perception of trust [59, 89]. Further, perceived trust can also affect user's choice and willingness to use different digital products and services [58, 68, 102, 111]. An in-depth discussion on psychological and social mechanisms of trust perception is beyond the scope of this work. Our work is based on two commonly agreeable features about trust: 1) trust varies by the subject and the object (including systems), and 2) there are different degrees of trust. More specifically, we allow each individual to express and exercise their own trust preferences, affecting the behaviour of the system. This incorporates additional dimensions not considered by existing literature on MPC, where the discussion of threat models as part of security is the main focus (see Section 2.3), and aligns well with our aim to support user autonomy. We refer to this as *individual-based, user-centric trust* (or simply *user-centric trust*), and discuss its implications in system security in Section 4.3.

2.2 Personal Data Stores (PDS)

Personal Data Stores (PDS), such as openPDS [32], Solid [70, 92], or Databox [78], champion a decentralised data paradigm by empowering users to retain control over their data within their own data stores, rather than having it locked away by large platforms, thus fostering user autonomy. While there are subtle technical and design differences between them, users generally store their data in a PDS, and applications make requests to the PDS to use (and store) data, with users making the final decisions about data usage. PDS also offers practical data protection controls, allowing users to establish preferences for data access, and to audit access to their data.

While PDS ensures granular control over data access and secure data transmission, privacy faces challenges once an application receives data, especially in scenarios involving collective data use. For instance, consider a use case which we will demonstrate later in Section 5.2.1: in a PDS-enabled context, gig workers such as Uber drivers would store their payroll data in their PDS (rather than locked away in their working platform's data storage, e.g. Uber's). Now they wish to perform a collective computation task, such as calculating their group's average salary for negotiating better pay rates. A typical solution would require them to grant read permission to their salary data to a designated individual (such as a worker or another delegate), who uses an application to read all the data and then compute the average salary. However, this involves raw data transfer[2] from each worker's PDS to the designated individual performing computation. Workers have no control once the data transfer is complete, and that individual can in principle perform arbitrary actions on those data. This issue persists **??** all PDS users, all types of data, and all types of collective computation, such as calculating census-like data statistics for a population, or deriving collective insights from health records, as mentioned earlier in Section 1.

Some PDS research offers mechanisms to ensure privacy in specific scenarios. For instance, openPDS includes subdivided personal data stores with access controls and local derivations, hosted locally or in the cloud [32]. Similarly,

[2] The network communication channel can be secured by standard means, such as SSL/TLS. However, the issue here is the original "raw" data is now copied to the recipient, that application.

Databox provides a combination of local and remote data stores configured to permit only authorised applications to access personal data [78]. While both emphasise the importance of privacy-preserving aggregation *within individual PDS*, they do not specifically test performing privacy-preserving data aggregation tasks *among a set of users*. Databox has been demonstrated to have the potential for distributed privacy-preserving machine learning with an implementation of Federated Learning (FL), orchestrated by a central server [110]; Meurisch et al. [75] supported a similar approach with the addition of using secure enclaves like Intel SGX [29] for the central server. However, such approaches face the same challenge of relying on one central server, and inherit properties and requirements for FL, thus being machine-learning-only and requiring redevelopment when the algorithm or model changes. While openPDS suggests the use of MPC for retrieving aggregated results from multiple PDS [32], but to the best of our knowledge, this has not been implemented or evaluated . On the contrary, [9] considered combining PDS with MPC impractical due to the large number of data providers. In general, a mechanism for privacy-preserving data access and processing is needed above and beyond the present features of PDS systems.

Solid [92] is a prominent PDS system renowned for its basis of Web standards and focus on interoperability, achieved by a modular architecture and a clear separation of roles. Solid allows federation with other users' PDSs, known as *Pods*, while preventing vendor lock-in of either PDS hosts or applications. In this work, we explore the implementation of decentralised privacy-preserving computation in a Solid-based architecture rather than Databox or OpenPDS due to its open and standard-based design. We also discuss how our approach can be adapted to other PDS systems.

2.3 Privacy-Preserving Mechanisms

Earlier in Section 2.1, we noted that lots of research in computer science focused on the data confidentiality aspect of privacy, which will be the main topic of this subsection. Other than that, there are also some research focusing on information flow management and/or contextual integrity, such as Information Flow Control [79, 85], various forms of data access/usage control [65, 108] (e.g. Data Terms of Use [109], ABAC [16, 52]), data governance rules [91], and contextual integrity modelling and application [30, 63]. They directly or indirectly relate to Nissenbaum's contextual integrity view of privacy [12, 80, 81], where privacy is considered to be the appropriate data usage and flow in a given context. We do not further detail them because of the difference with the scope of this work. They express or determine whether information usage should be permitted, based on perceived data usage principles; our work, on the other hand, provides a new means for data usage principles.

For privacy-preserving mechanisms, there exist several complementary approaches, including data modification, data minimisation, and data encryption [76].

Data modifications rely on obfuscation- or perturbation-based methods [3, 51] to alter or sanitise user data, preventing it from being linked to specific individuals. As a result, it often leads to a difficult trade-off between privacy and utility.

Data minimisation strategies aim to achieve optimal computational results by adjusting the computational model and reducing the volume of required data. Federated Learning (FL) is a machine-learning specific approach that trains models over distributed datasets while limiting data movement to central servers, transmitting only model updates [62, 103]. Meurisch et al. .

Data encryption methods utilise encrypted user data to ensure integrity and confidentiality during data sharing. There are two major data encryption approaches relevant to our work, homomorphic encryption (HE) [44] and (Secure) Multi-Party Computation (MPC) [86]. HE is able to analyse or manipulate encrypted data without revealing the data; however, it is limited by its low computational efficiency and limited operations. MPC encompasses a class of cryptographic protocols that rely on the secure evaluation of a function over sensitive data shared across multiple parties. MPC has the benefits of not losing precision and performing any type of computation, providing a promising option for privacy-preserving computation over dispersed data sources. We will briefly present some properties of MPC below, and discuss the challenge of using MPC in a decentralised (PDS) setting.

Secure Multi-Party Computation (MPC). Given an environment with n parties $P_1 \cdots P_n$, their corresponding inputs $x_i \cdots x_n$ and a function f, an MPC protocol computes $y = f(x_1 \cdots x_n)$ without revealing any input x_i to a party P_j $(i \neq j)$. Traditionally, two security notions have been considered for MPC [82] – *semi-honest security* and *malicious security*. Protocols with semi-honest security are relatively more efficient and protect against passive attackers that do not deviate from the protocol. In contrast, protocols with malicious security also provide security from attackers that may deviate from the protocol. The number of parties that an attacker could corrupt or compromise is another factor of security in the contexts involving multiple computation parties. Protocols assuming an *honest majority* ensure security under the assumption that fewer than half the parties could be corrupted by an attacker. In a *dishonest majority*, the same assumption does not hold. MPC protocols are generally realised using a combination of primitives such as oblivious transfer [87], garbled circuits (GC) [105] and secret sharing schemes [17, 94]. Here, we focus on additive [18] and Shamir secret sharing [94]. Understanding the main discussions and results in this paper does not require knowledge of the details of MPC; therefore, we omit them for brevity.

In prior work, MPC has been mostly explored in settings where each computation party has direct access to data (i.e. data providers are computation parties), about the security properties and the performance of protocols and frameworks [21, 55]. Work on combining MPC with distributed data sources partially addresses this issue: Mohassel and Zhang [77] proposed MPC-based protocols for specific AI algorithms, but they only experimented with distributing user data among two non-colluding servers; Bonawitz et al. [19] proposed an efficient model to securely perform FL over multiple users, but assumed a (single) trustworthy server as with FL in general; Rouhani et al. [90] use GC to securely perform scalable Deep Learning execution over distributed data from individuals, but is also constrained to the properties (e.g. performance) of GC; work on Private Set Intersection (PSI) like [43] and [2] involved many autonomous parties, with or without a central server's help, while intensively exploiting properties of PSI.

Despite their success in increasing the number of data sources, little attention has been paid to the providers of these computation parties and their relationship to the data subjects, especially for *general* MPC supporting a wide-range of computation tasks. Essentially, this implies a *centralised trust* of an organisation providing and choosing computing parties. We discuss in Section 3 how this exhibits problems when applying to a decentralised context such as PDS.

Differential Privacy (DP). Output privacy represents a distinct privacy aspect that complements the privacy protection provided by MPC. While MPC ensures that the inputs used in a computation remain undisclosed, output privacy goes a step further by preventing reverse inference based on the revealed computation results. Differential privacy, a formal mathematical concept introduced by Dwork [38], plays a crucial role in restricting the disclosure of private information contained in a database when employing a computation algorithm. In simpler terms, an algorithm is considered differentially private if an external observer, upon observing its output, remains unable to determine whether a specific individual's information was utilised during the computation process. In the paper, we show with an example how this complementary notion of privacy can be implemented in our proposed solution.

3 MPC in Decentralised Settings

3.1 Challenge in Decentralised PDS Context

Existing MPC literature places emphasis on the security-related assumptions about different computation parties, laying a necessary foundation for the Data Privacy (R1) requirement. However, there is little to no discussion on *who specifies the group of parties and the prescribed security properties.*

Usually, existing work focused on a platform-based setting (e.g. [14, 19]), so the security property is often assumed as a given because the *platform* will determine the parties and their rights on behalf of users. Thus, *trust* (and therefore security) is still centralised on the platform, despite using MPC. This contradicts the requirements of User-centric Trust (R3) and User Autonomy (R2), leading to a misguided privacy promise. This is the reason we call it the **'omniscient' view of security**, in which an entity 'knows' the security properties of all parties in advance.

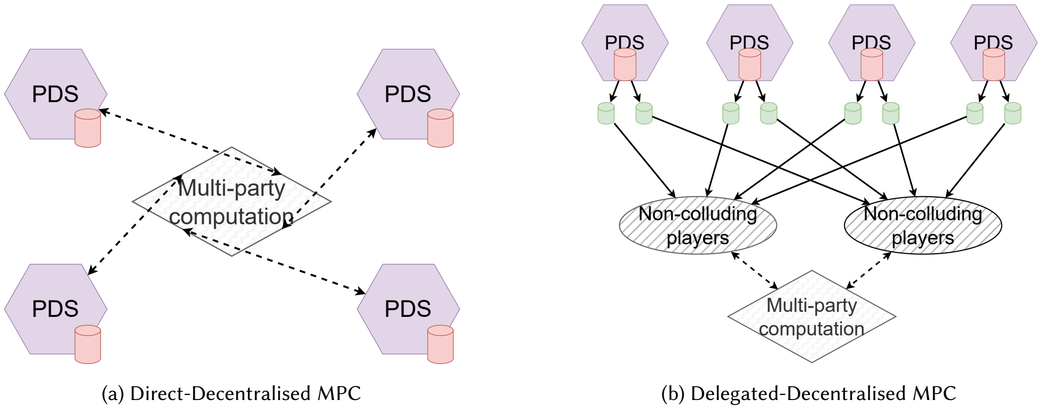
Nevertheless, their discussion led to a useful observation, which is the distinction between *data providers* and the *App (or App user)*: a **data provider** is someone (or someone's PDS) who contributes data to MPC computation; the **App** (or **App user**) is a person or entity who initiates / intends to perform the collective computation over multiple data providers' data. Naturally, the App user is not necessarily a data provider, and, more importantly, the App user does not necessarily represent the interests or preferences of *all* data providers. The pitfall of the said existing practices results from obfuscating these two roles by naively executing MPC while letting the MPC App developer(s) determine the computation facilities – by using the App user's machine and/or servers provided by the App developers. However, from the data providers' perspective, these computation parties will not be trustworthy as they can easily collude, unless the data provider fully trusts the App user. This is rarely the case and negates the need for MPC in the first instance.

In an ideal decentralised context with PDS, not only the data storage is decentralised, but also the central trusted party is removed, to ensure data privacy and individual autonomy over data use. Therefore, the platform as conceived in a centralised setting no longer exists, and the MPC App no longer predefines the set of computing facilities and their security features. Aligning with the ethos of decentralisation and PDS, it is the data owners/providers who should possess the autonomy and ability to administer their trust preferences, and control who has access to their data.

However, because different data providers have different preferences, it is inevitable that they will trust different actors for the relevant actions; also, because they only perceive trust from an individual perspective, one data provider's trust (of security properties) is not transferrable to another data provider. This creates a predicament for MPC because it necessitates a global view of security properties. Therefore, to enable MPC in such a setting, an alternative mechanism for establishing permissions is required, along with a mechanism for determining security properties and selecting computing facilities .

This is the key reason we propose a shift from an omniscient view of security properties (characteristic of platform-led settings) to an **individual-based, user-centric view of security properties**. We argue that the assumption that the system designer knows better than the individuals should be challenged, and individual's ability to make decisions for themselves should be respected. Collective security properties can then be determined and optimised based on individual views, provided that appropriate mechanisms are employed, as discussed later with Libertas. That answers three interweaving questions:

a) Who will carry out the computation; b) Why are these parties selected; c) What are their security properties?



(a) Direct-Decentralised MPC

(b) Delegated-Decentralised MPC

Fig. 1. MPC Models in decentralised settings. Data provider is denoted here as PDS.

3.2 Direct- and Delegated-Decentralised Models

Since our goal is to explore *how* to combine MPC with decentralised PDS, it is useful to revisit the models of MPC. In particular, we focus on *general* MPC protocols that support a wide range of computations, aiming to fulfill the generality (R5) requirement. Thus, the decentralised computation models for MPC can be classified into two broad categories:

(1) *Direct-Decentralised:* (Fig. 1a) In this setting, each *data provider* is a computation party (**player**), following an MPC protocol to carry out secure computation;

(2) *Delegated-Decentralised:* (Fig. 1b) In this setting, *data providers* are different from *players*. Data providers send secret shares of data to these players (such that no single player can make sense of the data independently). The *players* perform the main MPC computation between one another.

The direct-decentralised model reflects the traditional interpretation of MPC – each computation party (*player*) holds their data and performs computation. The data provider is the same as the player, and the only relationship to discuss is between different players. On the other hand, the delegated-decentralised model separates the computation parties and the data providers, leading to different relationships.

3.3 Utilising MPC on PDS

With this observation, one possible approach is to follow the direct-decentralised model: use the PDS (or a dedicated server for each of them, same below) as the computation parties in MPC. Because each data provider trusts its own PDS, at the minimum a dishonest-majority protocol can be used. Further, if an appropriate punishment or incentive mechanism exists (e.g., retaining social relationships with peers or blacklists), data providers would want to maintain their reputation, and therefore can generally form an honest-majority group.

However, beside the technical requirements (the capability to accept custom computation in PDS), this also involves a performance issue – the number of data providers (thus *players*) could be very large. As we show in Section 5.1, this scales poorly. That is why we would like to avoid this model.

Another approach uses the delegated-decentralised model. The secret sharing of data between the data providers and the computation parties (players) can be realised using a mechanism like the "*client*" approach in [31], which can handle corrupted participants. For *players*, we can expect internal or external services providing agents as player candidates. This provides the necessary background design, but the question about trust remains: how do we determine the trust relationship(s) between the data providers and those computation facilities? Again, we cannot allow the App developer to define which facilities to use; therefore, we must incorporate data providers' preferences.

As we will discuss next, in Libertas, by allowing users to explicitly express their trust preferences, it is possible to dynamically select computation facilities while respecting users' autonomy. In particular, as we will discuss later in Section 4.3, a subset of the trusted agents of all data providers can form a non-colluding or honest-majority group (e.g., intersection of trusted agents). This forms the possible basic for faster MPC computation.

The delegated-decentralised model is also expected to scale better than the direct-decentralised model because fewer *players* participate in the computation, which is verified by our benchmark (Section 5.1). With this in mind, the next section explains the Libertas architecture we have developed for employing delegated-decentralised MPC with PDS. This approach allows for the utilisation of data providers' preferences for trusted participants, while maintaining compatibility with existing protocols.

4 Libertas: Architecture for Privacy-Preserving Decentralised Collaborative Computation

To address the aforementioned challenges and requirements, we propose an extended architecture based on Solid [92], called **Libertas**, illustrated in Figure 2. Our architecture leverages existing mechanisms such as authorisation and access controls, and is compatible with underlying protocols. Although we demonstrate a Solid-based architecture, we will also briefly discuss how the proposed architecture can be adapted to other PDSs, thanks to the modularity of the design.

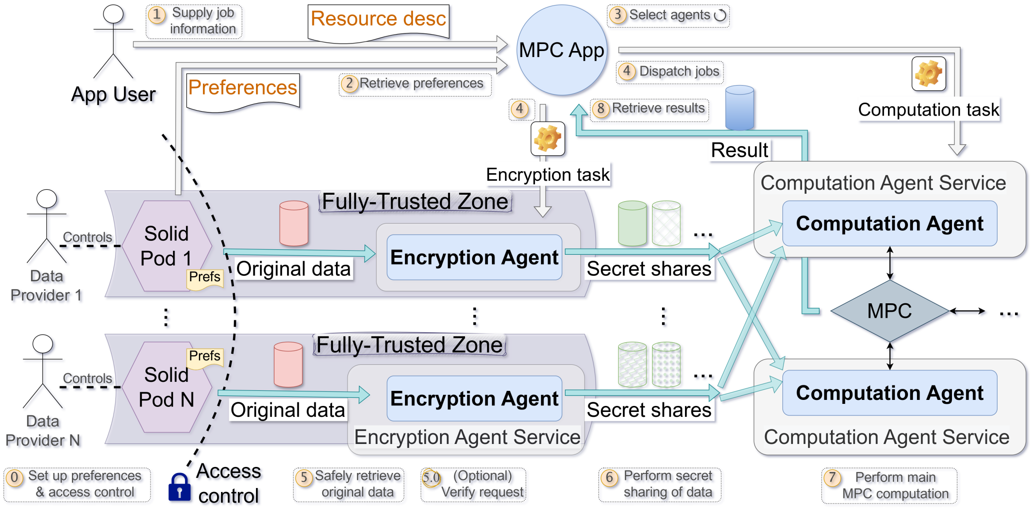


Fig. 2. Libertas: A modular architecture for meaningfully integrating MPC with Solid. Three dots denote possibly more repetitions. Architectural core components are coloured.

4.1 Architecture Design

Core components. The core components are coloured in Figure 2. Users (data providers) store their data in their own Solid Pods. The MPC App initiates the computation and dispatches relevant tasks to the agents who carry out further computation. The **encryption agents**, acting as the *clients* [31] of MPC, retrieve data from Pods (Step 5), and send secret shares of data to the computation agents (Step 6). These **computation agents**, acting as the *players* of MPC, execute the primary MPC computation amongst one another (Step 7). Finally, the MPC App obtains the results (Step 8). This process differs slightly from the depiction in Figure 1b, as Solid Pods lack computational capability. Therefore, in addition to the PDS, *encryption agents* are explicitly introduced to act as the clients.

Data-provider-centric configuration (Step 0). Prior to any computation, each data provider creates a **preference** file, containing their trust preferences regarding the list of trusted actors, especially the encryption and computation agents.[3] More precisely, it contains the *services* providing such agents, which will be elucidated later in the discussion on *separation of duty*. As files are protected by Solid's access control mechanism (Web Access Control [15]), the data provider must also grant relevant permissions. By doing so, the provider makes the preference file readable to the MPC App or the App user[4] and grants the trusted encryption agents (identified by their WebIDs [101]) read access to their data. These steps only need to be performed once and can remain the same for all future MPC tasks if the preference does not change. If the data provider chooses to revoke trust from anyone, they can simply withdraw access permissions from their Solid Pod and/or removing them from the preference file, as appropriate.

App usage (Step 1 & 4). The App user submits a **resource description**, containing a list of data (resources) and the relevant *preferences* from each data provider. Assuming the appropriate permissions are granted, the MPC App reads the preference files, identifies the encryption and computation agents (to be discussed later), dispatches relevant directives (and MPC tasks) to them, awaits the completion of the computation, and retrieves the result.

The **encryption task** describes how to secretly share data with each player, while the **computation task** specifies how to execute the main computation among the computation agents. Both the encryption task and the computation task should be accompanied by the list of computation agents (instances of such agents obtained from the services) to facilitate connection establishment. We utilise MP-SPDZ [55] as the MPC framework, leveraging its supported *client* mechanism, which is based on a variant of the SPDZ protocol [31], for the secret sharing of data. In our implementation, the tasks are transmitted as source code, and the agents will (compile and) execute them upon receipt. Additionally, the App can customise various aspects such as the number of computation agents (and methods to determine them), MPC protocol, parameters, etc., as MP-SPDZ supports a wide range of protocols and parameters.

Trust and agent selection (Step 2 & 3). The preference files provided by data providers serve as the primary source for the MPC App to select relevant agents (and for encryption agents for further verification, for additional security). In our prototype, each preference file includes a list of trusted encryption agents and a separate list of trusted computation agents.

It is important to note that the trust requirements for encryption agents and computation agents differ: encryption agents must be fully trustworthy from the data provider's perspective, while computation agents only need to be semi-honest and non-colluding. This distinction arises because encryption agents have access to raw data, whereas computation agents receive only secret shares of data. Ensuring the non-collusion of computation agents is crucial: if all of them collude (or a significant portion of them collude, depending on the protocol), they can combine their shares and reconstruct the original data. These agent services can be provided by public or private entities. For instance, the PDS host may deploy an encryption agent service alongside the PDS service itself, making it relatively straightforward to establish trust with the data providers.

[3] Further discussion on principles regarding trust and agent selection is provided in the subsequent section. Additional preference files associated with data resources are used for additional security guarantee, also discussed later.

[4] Because at each time, the App and App user authenticate themselves together. For simplicity, we will only refer to them as the MPC App.

Choosing the encryption agent for each data provider is straightforward for the MPC App: simply select one of the trusted agents specified by that data provider. It is permissible to choose the same encryption agent (service) for different data providers. However, selecting computation agents requires more careful consideration. Various approaches can be adopted for this purpose, and we have implemented two extreme versions: taking (a subset of) the intersection of the trusted computation agents of all data providers, and taking the union. Further details are discussed in Section 4.3.

4.2 Additional Properties

Separation of duty. The Libertas architecture effectively partitions the responsibilities of various stakeholders, resulting in reusable components across multiple tasks. Both types of agents, encryption and computation, can be automatically repurposed by different MPC applications because they solely provide computing (and data fetching) *facility* rather than *logic*. This suggests the existence of public or private, free or paid, agent services from which data providers can make selections. The actual computation is instantiated by the MPC App, which furnishes the logic (and parameters) for the computation. Because the agent services operate independently of the App, each App can seamlessly utilise existing agents (agent services) specified in the data providers' preferences.

Furthermore, as the App user may not necessarily be a data provider, this setup affords data providers the flexibility to authorise third parties to utilise the data for legitimate computations, such as platforms, worker unions, or government entities. Access controls are implemented to prevent unauthorised access, and the security mechanisms and secure protocols of MPC safeguard the raw data from exposure to the App user.

App verification (Step 5.0). The encryption agent may additionally perform *App verification* to enhance the security properties of the system. To support this, the data provider must express the list of trusted Apps in a resource (called *trusted actors*), making it readable to trusted encryption agents, and refer to the URI of *trusted actors* in the *description resource* of the data [26]. Upon receiving computation request, the encryption agent retrieves the *description resource* and then the *trusted actors*; it also attests the identity of the App by checking the request headers (following Solid-OIDC [27]); then, it verifies if the App is within the *trusted actors*. This process both verifies the real identity of the App, and also verifies the App is within the trusted actors.

Variants. As outlined earlier, the Libertas architecture seamlessly integrates with the existing Solid protocol and leverages its established mechanisms. From a broader perspective, this architecture is not confined to a specific ecosystem or framework for either the PDS or MPC framework, but rather relies on functional properties such as the presence of an access control mechanism. This flexibility allows for its adaptation across various systems.

To accommodate different (PDS) systems, the different modules of Libertas may be merged with them, yielding derived architectures with distinct trade-offs. For instance, merging the encryption agent with the data store (PDS) simplifies the user's task of identifying and assigning trust to encryption agents. However, this approach may complicate the PDS service, potentially deviating from its original protocols. This exploration could be particularly relevant for PDS systems with computation capabilities, like openPDS. Alternatively, integrating computation agents with the PDS services and randomly sampling a subset of them via the MPC App could be another avenue to explore.

Moreover, the flexibility extends to the MPC framework, allowing for seamless substitution while preserving the architecture's integrity.

4.3 Threat Model

In Libertas, four major parties play essential roles: the Data Provider (DP), the App (including App user), the Encryption Agent (EA), and the Computation Agent (CA). This presents a distinct threat dynamics compared to existing MPC literature, where only EA and/or CA are considered (Section 2.3). This section discusses that in depth.

4.3.1 Assumptions. As a basis for discussion, here are the general assumptions:

(1) All network transmission is secure, for example through SSL/TLS.
(2) Output data from Libertas are not a source of privacy concerns (e.g. through Differential Privacy as discussed in evaluation).
(3) User's PDS/Pod is fully trustworthy.
(4) The implementation and theory of the authorisation and authentication mechanisms come without flaws.
(5) Hosting machine and software stack do not involve additional security concerns.
(6) Each party in the framework has a unique identifier to be identified.

Besides that, since the motivation for Libertas is to support user autonomy (while also achieving scalable collective computation), we also assume the DPs are semi-honest and non-colluding, and they make sensible decisions (e.g. to their preference documents and access control settings) to the best of their knowledge. That means, DPs may choose trusted agents for their own benefits but not others.

It is worth emphasising that Libertas utilises individual-based, user-centric trust: each DP only makes decisions for himself/herself, and they do not need to know or consider the decisions of other DPs. This individual-based view is a key factor for the discussion below, as it is the only source of information in a user-autonomous setting. We will not repeat this at all times in the discussion below.

4.3.2 Threat from Encryption Agent. An EA can access all data that it has been granted read permission to; therefore, we require every EA to be fully trustworthy. This is why it may be impeding to combine the EA with the PDS service, as discussed above in *variants* (see also discussion below about the potential threat from the App). If an EA is compromised, all data it has access to are at the risk of privacy leak. The only countermeasure relies on avoiding the EAs from being able to enumerate the resources in users' Pods, such as by denying access from trusted EAs from a parent/ancestor container (but allowing access to the data resource file). However, that only protects data that have not been accessed by the EA.

As we have employed access control to the data resources, untrusted EAs will not be able to gain access to the data.

4.3.3 Threat from App. As an App will dispatch tasks to EAs and CAs, and is responsible for the selection of EAs and CAs, we require it to be semi-honest. That means the App will correctly follow the protocol, including retrieving preference files, selecting EAs and CAs, and dispatching tasks. Since the App has no access to raw data, and we have assumed the computation result is not sensitive, there will be no threat concern for a semi-honest App to obtain sensitive data, such as the raw data of each DP.

An untrusted App may send random requests (dispatching tasks) to EAs, in the hope that it may opportunistically choose the right combination of data resource and EAs. In this case, it will set up and instruct EAs to perform secret sharing to contaminated CAs controlled by itself (see also the discussion of threats from CAs below). In this case, the *App verification* procedure discussed in Section 4.2 can eliminate such threats, as the EA receiving the request will be able to identify that the App is not trusted by the data provider for this specific resource, and ignore the request.

For this reason, the vanilla variant of Libertas prefers EAs to be provided separately from PDS services, as this reduces the possibility of an untrusted App choosing a sensible combination of data resources (, data providers) and EAs on the Internet, by a factor of the possible EA services on the Internet. However, we also proposed the *App verification* mechanism for variants that combine EAs with PDS to simplify the trust selection (given that both PDS and EA share the same trustworthy requirement), or implementers who prefer a more cautious approach.

Threat from compromised App. If a trusted App becomes malicious, it may perform one of the following to form a threat to data privacy:

(1) Send malicious computation tasks to the CAs (e.g. revealing input directly as output);
(2) Instruct EAs to use malicious CAs for computation, and gain access to input from these CAs (this includes selecting MPC protocols with inappropriate security guarantees).

For case 1, the DP can assign trustworthiness to computation tasks, such as by digital signatures or verifiable credentials, and share that with the App in advance. The CA will need to verify them upon receiving the task. For example, when sending the computation job (to the CAs), the App must also send the digital signatures of all DPs (confirming approval) for this job (code); the EA will send the public key of the DP to the CAs; each CA verifies all the signatures are valid, and then proceed with computation. Note the assumption that (at least one) CAs are trustworthy, as to be discussed in the next section.

For case 2, a countermeasure is to verify the appropriateness of the chosen MPC protocol and the chosen CAs. As discussed in the next section, not all chosen CAs need be trusted by the DP, but there is an appropriateness between the (trustworthiness of) chosen CAs and the MPC protocol. Since CAs will be established communication from EAs and thus named in the job dispatch from the App, the EA (of each individual DP) can verify the trustworthiness of the chosen CAs. For this, the EA will need to 1) verify the amount of chosen CAs that are trusted by the DP, 2) calculate the allowed MPC protocols[5], and 3) send that to the CAs; each CA will 4) verify if the chosen MPC protocol is within the list, and abort if otherwise. This works on the assumption that at least one chosen CA is among the DP's trusted CAs; if otherwise and that is a concern (see Section 4.3.5 for a scenario where this may not be a concern), the EA can abort the computation immediately upon discovering this (at step 1).

4.3.4 Threat from Computation Agent. We require the CAs to be semi-honest and non-colluding *to the DP nominating them in the preference file*. However, unlike EAs, CAs are shared across all DPs for each computation run, making their security properties more complex.

The computation run will need to choose an MPC protocol with appropriate security guarantees given the security properties of the chosen agents to avoid data privacy risks. While stronger protocols offer theoretically higher privacy guarantees, that is unnecessary for performance reasons. The chosen protocol should have the same security assumption as the security properties of the chosen agents. Because Libertas uses the *client* mechanism [31] for secret-share of data from EAs to CAs, privacy is protected even if all clients (EAs) are corrupted and all-but-one players (CAs) are corrupted. In the meantime, if all CAs are corrupted, no privacy is guaranteed between CAs (not considering EAs) regardless of the chosen MPC protocol. Thus, the secret-share mechanism will not be the privacy bottleneck, and we will not discuss it below.

[5] The security property of each MPC protocol is well-known public information, so the calculation here is to simply select the set of MPC protocols that fits with the security property fact – the ratio between trusted CAs and chosen CAs. To strengthen security or further restrict the range, the DP may specify a list of allowed MPC protocols, and the EA calculates from this set.

For agent selection, in the best case, if the agent selection algorithm is *subset*, which takes the intersection of trusted CAs from the preference files of all DPs, the chosen CAs will be semi-honest and non-colluding to all DPs. In this case, employing a semi-honest MPC protocol will suffice, such as Shamir.

Alternative mechanisms are needed when an insufficient number (but still some) of suitable CAs are found in the subset, because non-common CAs are not necessarily semi-honest or non-colluding with respect to the other DPs. In these cases, to accommodate the required number of CAs, the selection algorithm will need to choose enough number of CAs from the non-intersecting union of CAs, leading to non-trusted CAs in the final list of chosen CAs. Therefore, with the increase of non-trusted CAs, MPC protocols with stronger security guarantees may be needed, up until a covert (e.g. ChaiGear or CowGear [57]) or malicious (e.g. MASCOT [56]) protocol, to detect (and abort) and/or tolerate dishonest behaviours in this context, at the cost of performance. For example, if only one common CA exists, we can use protocols like MASCOT for dishonest-majority security.

4.3.5 Special Case for Lack of Commonly Trusted Computation Agents. In the worst case, no common trusted CAs exist, and thus each individual DP cannot get assured a security property of the chosen CAs. In that case, either the computation is not possible, or the DPs have to accept a risk. We note a special case where an acceptable slight risk is exhibited to perform the computation in this scenario: if we *randomly* select from the union of all trusted CAs of every DP, and we use a malicious-majority protocol, such as MASCOT, privacy can still be guaranteed provided that not all the computation agents are corrupted.

Assuming each DP provides the same number of distinct CAs, this risk probability is $P_{risk} = \prod_{i=0}^{m-1} \frac{k-i}{n-i} < (\frac{k}{n})^m$ for random choice, where n is the number of agents in the union of trusted CAs of all DPs, k is the number of corrupted agents in the union and m is the number of chosen CAs. With a sufficiently large group of DPs (thus large n) and a limited k, this probability is close to zero. This implies that as long as the majority of DPs are honest, the risk is very low and is acceptable if possibility of computation is more important.

Of course, future work can explore alternative selection algorithms for better suitability in different scenarios, especially utilising context-specific information, for an efficient balance between security and performance.

5 Empirical Evaluation

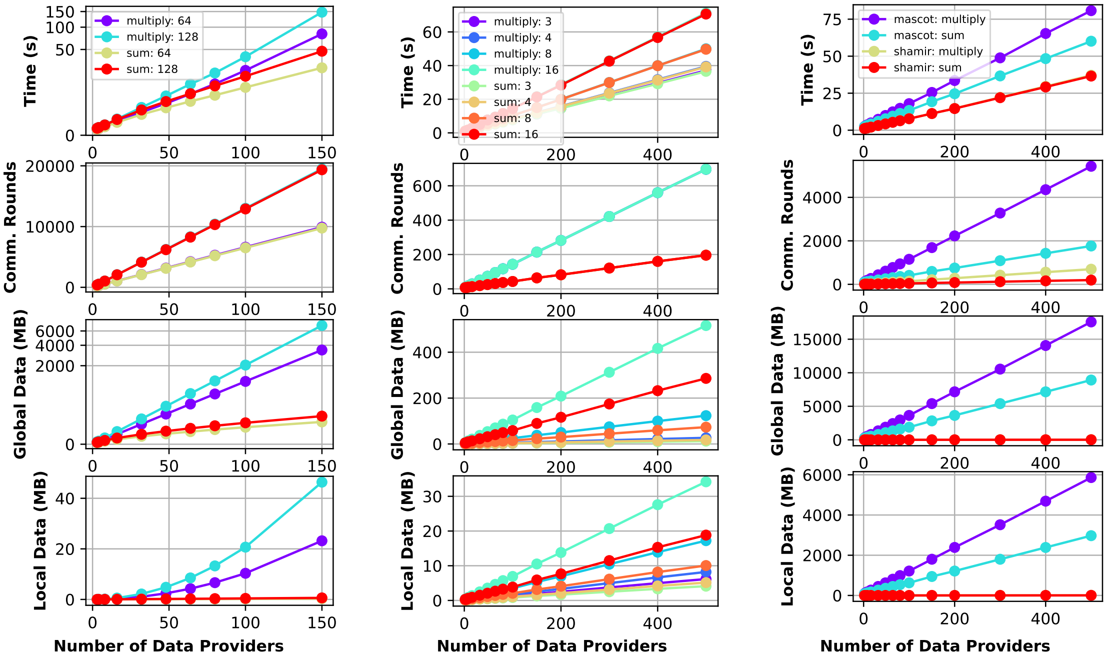
In this section, we present a series of benchmarks on various tasks. Since our work fills a gap in the application of MPC rather than proposing new MPC protocols, and we utilise an off-the-shelf MPC framework, comparing performance against other works as a baseline may not be meaningful. Instead, we concentrate on scalability patterns concerning a distinctive feature in the decentralised PDS context: the number of data providers.

5.1 Scalability of the Two MPC Models

Using the same framework, MP-SDPZ [55], we conducted a scalability comparison between two MPC models: direct-decentralised MPC and delegated-decentralised MPC.

In the benchmarks, each data provider possesses an array of data and employs MPC to compute an element-wise operation across all data, culminating in the summation of these results.[6] Our benchmarks encompass various computation operations (such as sum and multiplication), parameters (such as array sizes and numbers of parties), and protocols (such as Shamir and MASCOT) in the two models, utilising a server with 2x 8-core Intel E5-2650v2

[6] For instance, in a two-player setting with multiplication as the operation, this is akin to computing the dot-product.



(a) Direct-decentralised MPC. Time and global data are in **cubic-root scale**. Legend shows the computation & array size.

(b) Delegated-decentralised MPC. Legend shows the computation and number of players. Array size is fixed to 128.

(c) Protocol comparison in delegated-decentralised MPC, with 3 players and array size 128. Legend shows protocol & computation.

Fig. 3. Benchmark of two MPC models in different settings.

(2.6GHz) CPUs and 48GB RAM. The primary factors/metrics recorded include time, rounds of communications, and data transmission.

The results are depicted in Figure 3. In summary, we observed and confirmed the following:

- In all settings, computation cost of delegated-decentralised MPC grows linearly with the number of data providers, while that of direct-decentralised MPC grows polynomially.[7]
- Cost for more complex computation (multiplication) grows faster than simpler computation (sum). Sometimes this is in magnitudes of difference (i.e. multiplication vs sum for direct-decentralised).
- Protocols with stronger security guarantees require significantly more resources, and are more costly than more complex computational operations. For example, sum operation in MASCOT is more costly than multiplication in Shamir.

From the results, we conclude that the delegated-decentralised model is more appropriate for a significant decentralised context such as Solid. This can be intuitively explained: increasing the number of data providers equals to increasing players in a direct-decentralised model, thus leading to more rounds of transmissions, and a higher number of communications and data transmission in each round; while more data providers in a delegated-decentralised model results in more clients, but does not change the number of players inter-communicating to perform the main computation.

[7] Quadratically for sum; cubically for multiplication.

We also observe that each server can handle a sufficient number of players/clients, without impacting performance. Although we did not specifically test the maximum capacity of the server hardware, we stopped at the shown numbers (150 players for direct-decentralised and 500 clients for delegated-decentralised) because of soft operating system limits (esp. number of file descriptors). The shown results demonstrate the feasibility of running dedicated servers, for the *agents* proposed in the Libertas architecture.

Furthermore, in our benchmark, the time required for even basic operations cannot be ignored, especially for protocols with stronger security guarantees. This may pose challenges for tasks requiring real-time processing in large decentralised contexts. However, it is important to note that we did not fine-tune parameters, and the time includes initialisation and connection establishment between all parties, which can impact the reported performance. Further research in MPC may lead to performance improvements in the future, which can be directly integrated into Libertas, such as a new protocol. Conversely, for tasks that do not require real-time processing, this is less of a concern, particularly for tasks where privacy outweighs efficiency, such as medical records or highly sensitive financial data. The upcoming use cases discussed below provide additional insights in this regard.

5.2 Libertas Evaluation in Real-World Use-Cases

To demonstrate how Libertas can deliver impact, we assess its ability to support realistic use cases, such as gig worker empowerment, and differentially private synthetic dataset generation from decentralised data sources.

5.2.1 Gig Worker Empowerment. Gig workers are vulnerable to unfair treatment by their work platforms [22, 93]. They could benefit from gaining control over their own data and conducting statistical analysis on data aggregation, a practice repeatedly mentioned in literature [23, 24, 106]. They also informally perform this through unsecured social media channels and private text groups, demanding substantial effort from moderators and entail placing significant trust in fellow group members or platforms, also risking the exposure of sensitive information [106]. Several recent studies have highlighted that the preservation of workers' personal privacy is critical for their uptake of any new paradigm, as sharing salary information or work patterns may jeopardise their earnings or competitiveness amongst other workers [84, 107].

The first evaluation we present discusses how Libertas can facilitate gig workers by providing a more flexible, agile and privacy-preserving framework to empower the gig workers, without involving a single source of trust, reducing risks in existing approaches discussed above. We first present how Libertas can support the practice, including how the requirements of Libertas can be achieved by real-world power dynamics, and then present the technical solution aspect, especially the experimental conditions.

New Norm with Libertas. By adopting Libertas, gig workers will store each individual's data in their own Pods, and execute the computation by themselves or delegates with *computation rights* (but not *data access rights*) to the data. The fact that they may already trust a data intermediary can accelerate the adoption, because the original actors such as data intermediaries can still provide multiple functions, such as providing Pod storage, data analysis, and agents for computation. This may seem unnecessary if believing everyone to trust the same party, but immediately becomes useful if otherwise: because of the clear and agile design of Libertas, any single gig worker can choose different providers for each part, without necessarily impeding the collective computation, which is extremely difficult in existing practices.

For example, a gig worker may choose to: 1) store the data in a Pod at the work platform; 2) use a self-hosted EA; 3) permit a worker union for the analysis App; and/or 4) use CAs provided by different institutions, according to their individual preferences. In particular, real-world power dynamics can lead to sensible choices of such CAs, where different institutions have an intrinsic tension that prevents them from cooperating with one another. For example, workers might choose one CA from the public sector (e.g. *transportation authority*), one CA from their platform (e.g. *Uber*), and one CA from a worker union or data intermediary (e.g. *App Drivers & Couriers Union*). Their well-known reputation and structural tension result in a high likelihood for the majority of workers to trust them as CAs; even if not all workers trust the same CAs, as discussed previously, the computation can still be performed without privacy concerns.

Experimental Design. We present an experiment to assess the technical feasibility and scalability pattern of Libertas in such a context. As a use case, it tackles the wage discrimination often faced by gig workers [37], by providing insights of average wage across all gig workers from individually owned private income data. In the experiments, we assume that each worker is equipped with a Pod, storing their hourly income (simulated by assigning random values), and they have chosen appropriate actors for the collective computation using Libertas. We implemented an "average wage" MPC circuit in MP-SPDZ, calculating the average wage from individual earnings stored in their Pods. Experiments were conducted with varying numbers of data providers (10 - 1000) to discern the scalability pattern.

5.2.2 Synthetic Dataset Generation with Differential Privacy. As discussed in Section 2.3, while MPC ensures that the inputs used in a computation remain undisclosed, no privacy protection is provided for the computation results. Orthogonally, Differential Privacy [39, 40] provides a series of mechanisms to protect the output from being used to identify information about input. It complements the input privacy of MPC, and thus enables an interesting use case for Libertas for synthetic data generation. In this use case, we delve further by discussing the benefits and technical aspects of performing differential privacy on Libertas.

Formally, differential privacy provides a formal specification for a randomised function $\mathcal{M}$ that takes input D (from an input space $\mathcal{D}$) and produces output O (from space $\mathcal{O}$, not necessarily different from $\mathcal{D}$) – the function $\mathcal{M}$ is said to provide ϵ-Differential Privacy if for all *neighbouring* datasets[8] (of D) $D' \in \mathcal{D}$ and for all subsets $\mathcal{S} \subseteq \mathcal{O}$, such that

$$Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\epsilon} \cdot Pr[\mathcal{M}(D') \in \mathcal{S}]$$

To put it differently, the function, or algorithm, $\mathcal{M}$ produces indistinguishable outputs for two datasets that only differ by a single entry, bounded by the privacy parameter ϵ. Consequently, it safeguards against using the output to deduce whether a specific data record exists in the dataset or if a particular data provider contributed to the computation (assuming each provider contributes only one data record).

Differential privacy mechanisms can be used to generate synthetic data that mimic the statistical properties of real-world data while minimising privacy risks, offering a means to balance the need for data-driven insights and open availability of data with the imperative of protecting individual privacy [53]. Beyond privacy, synthetic data approaches are also being actively explored to overcome the limitations and shortcomings of real data for building more robust and fair artificial intelligence [41, 66]. Note that this is not to imply that differentially private synthetic data is a silver bullet for data privacy (let aside the differences in scope of privacy, as discussed in Section 2.1). Indeed, there are several known limitations of synthetic data due to the fundamental trade-off between accuracy and privacy risks [97]. But depending on the use case and context, such techniques can be very useful [20, 25, 42, 54] in practice.

However, having access to raw data to generate high quality synthetic data is often a challenge, partially because of the common assumption of the existence of a central curator, in most synthetic data generation approaches. This is similar to the gig worker empowerment scenario, where the impractical assumption of requiring a diverse set of

[8] Neighbouring dataset means D and D' differ in at most one entry, e.g. D has one more entry than D'.

contributors to trust a common central curator may lead to reduced or even dishonest contributions, consequently lowering the quality of data [72].

Using Libertas to coordinate the data providers provides a novel paradigm of synthetic data generation. To use Libertas, users (data providers) need only express the relevant trust information and assign access control in their Pods, all on an individual basis; and the architecture automatically guarantees the rest. It offers several benefits:

- Data providers do not need to coordinate with each other in advance or simultaneously with a single data curator, lowering the precondition requirements of trust obtaining and assignment.
- By ensuring privacy protection at both the input and output ends, users would feel more confident participating in synthetic data generation. This can lead to the creation of higher-quality and privacy-friendly open synthetic data sets for the common good.
- The use of costly MPC is minimised, as it is employed only **once** during the generation of synthetic data. Subsequently, the synthetic data can be used for running queries and analysis in a privacy-friendly manner **without** the continuous need for MPC . This approach is significantly more scalable than employing costly MPC for every query and analysis on sensitive personal data stored in personal data stores. Additionally, different synthetic data generation algorithms provide certain guarantees on the accuracy of queries, enhancing usability of the data.

Experimental Design. We implemented the classic Multiplicative Weights and Exponential Mechanism (MWEM) algorithm [47] (see Appendix 6 for more details of MWEM algorithm) for differential privacy as an MPC circuit,[9] and initiated computation through our prototype App. In the experiments, randomly sampled data and preferences are distributed across various resources under different containers, simulating different Pods within the decentralised architecture.

We conducted experiments under the following settings with varied number of data providers from 10 to 1000: (**Setting 1**) Fixed total number of data points (10,000), evenly distributed among data providers; (**Setting 2**) Fixed number of data points per provider (100). The MPC MWEM circuit transformed the data into a histogram using a fixed number of bins (10). The MWEM algorithm was executed with 60 randomly pre-generated queries for 30 iterations ($T = 30$) and an epsilon value of 1 ($\epsilon = 1$). Additionally, we conducted another set of benchmarks with a simple yet proven efficient optimisation (**Setting 3**): allowing clients to create local histograms and send these histograms instead of having players create histograms.

5.2.3 Experimental Settings. For both scenarios, we maintained the following common settings:

- Deployment of 3 computation agent servers and 1 encryption agent server, interconnected over a (virtual) LAN.
- Each computation agent resides on its own server, aligning with the non-colluding requirement discussed earlier.
- The agents utilise self-signed SSL certificates to ensure secure data transmission between them.
- All servers are equipped with 2x Quad-core Intel E5520 (2.2GHz) CPUs and 12GB of RAM.
- Data is hosted on a Solid server running Community Solid Server implementation [1], situated in the same data centre as the agent servers but not directly linked via (virtual) LAN.
- We employed the Shamir protocol, utilising the default parameters from MP-SPDZ.

[9] We implemented MWEM of 1-D dataset (integers), taking the final distribution as output, and do not perform mini-iterations during the multiplicative weights update step. With the produced distribution, one can sample a synthetic dataset.

We recorded the relevant factors for computation on the 1st player. Specifically, we noted the time for the entire job (*full-time*),[10] as well as the time after all connections were established (*comp-time*).[11] Each task was repeated at least 10 times to obtain an average result, thereby mitigating the impact of random fluctuations.

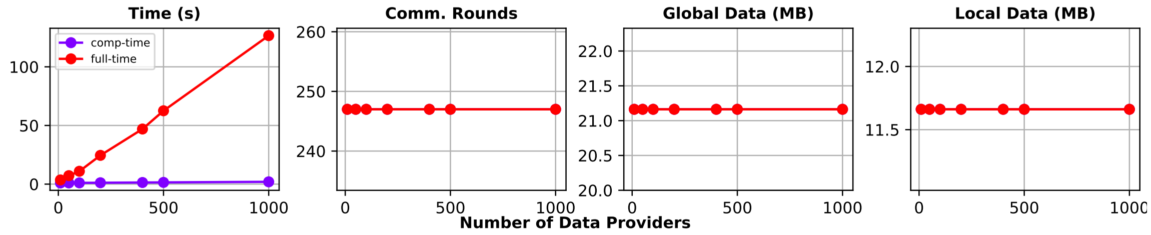


Fig. 4. Results for average wage computation in Libertas in the gig workers scenario.

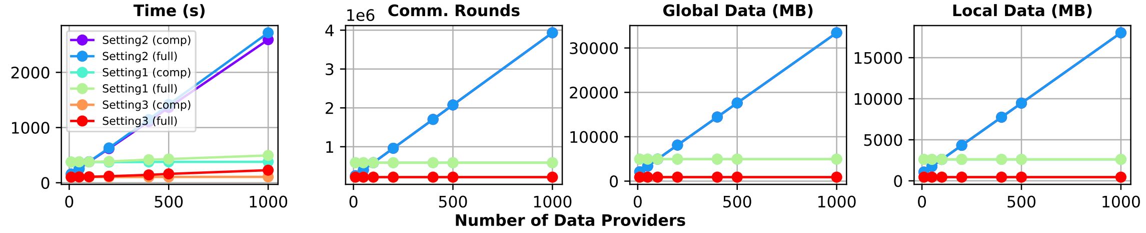


Fig. 5. Results for differentially-private synthetic data generation (MWEM) computation in Libertas.

5.2.4 Results. Figures 4 and 5 display the experimental results across various settings. It is evident that all factors exhibit a linear growth trend with the increase in the number of data providers. This aligns with the linear trend observed in the platform-agnostic benchmark for delegated-decentralised MPC (Section 5.1), affirming the overall scalability of the architecture.

Zooming into Figure 5, a comparison between Setting 1 (green lines) and Setting 2 (blue lines) reveals a notable impact of the total data amount on overall computation cost. Conversely, the red lines demonstrate that the resource requirement for Setting 3 (client-binning optimisation) decreased even further compared to Setting 1. This illustrates that a naive implementation may encounter performance bottlenecks with a large number of data providers, particularly during histogram creation from raw data. However, simple optimisations like client-binning can markedly reduce costs.

Furthermore, we notice that the full-time (time from the beginning until the end of computation) increases more rapidly than the comp-time (time for the main computation only). This disparity, termed as the "setup time", expands with the number of data providers due to the additional time required for encryption agents to prepare and establish connections with computation agents. This discrepancy could stem from factors such as Pod performance, network conditions, operating system constraints, or implementation specifics of MP-SPDZ. By isolating the compute time from setup time, we observe a gradual and slow growth in time with the number of data providers for both Setting 1 (e.g., 374.9s for 10 data providers versus 379.4s for 1000 data providers) and Setting 3 (e.g., 101.6s for 10 data providers versus

[10] To be precise, this duration spans from when the players/CAs connected to each other until the completion of the computation.

[11] This excludes the time required for encryption agents to prepare and download data, and to establish connections with computation agents.

106.6s for 1000 data providers); this also holds for other factors. This slow growth trend in computation time underscores the promising scalability of Libertas, as setup time optimisation can occur independently of MPC computation.

In real-world deployments, the network conditions between computation agents will inevitably impact performance. Task-specific optimisations, like the one demonstrated with client-binning, can substantially enhance performance. Moreover, there is potential for further optimisation by considering the exact computations involved (such as the MWEM algorithm), refining the code, and optimising the MPC protocol to streamline communication rounds and data transmission. We believe this is a sensible expectation in production environments.

The gig worker empowerment scenario exhibited a similar growth trend (Figure 4). However, a key distinction is that the setup time becomes dominant when the number of data providers is large. This underscores the importance of considering setup costs, particularly for relatively simple computations.

Our evaluation demonstrates that the proposed framework exhibits good scalability, with computation costs scaling linearly in relation to the number of data providers, consistent with our findings in the platform-agnostic benchmark. Moreover, the framework shows potential for significant optimisation based on the specific computational task at hand. Overall, these results provide a promising demonstration of the technical feasibility and scalability of Libertas's implementation with Solid, the possiblity of the proposed agent services, as well as Libertas's applicability in various computation tasks.

In addition to the empirical results, the case studies also revealed how real-life power dynamics can facilitate the assignment of trust preferences by data providers, and also proposed a novel paradigm for large-scale collaborative data utilisation while preserving input- and output-privacy. That provided confidence for Libertas to create real-world benefits on high-impact use scenarios. Relatedly, we note that privacy assurance and accommodating trust preferences are not only valuable from an ethical perspective but also from the perspective of encouraging adoption and delivering value, a topic for future studies to explore.

6 Conclusions

User autonomy, decentralisation, privacy-preserving computation, and collaborative computation are all desirable properties for a wide range of tasks and scenarios, but they may seem contradictory at a first glance for one system to achieve. In this paper, we addressed various challenges associated with building an end-to-end solution for this problem. We discussed the conceptual pitfalls of existing multi-party computation practices, and innovatively proposed an architecture featuring individual-based, user-centric trust to overcome such issues in the decentralised context. We presented a novel architecture called Libertas, which integrates privacy-preserving computation mechanisms like secure multi-party computation (R1 & R5) with personal data stores (PDS) in a modular fashion. Our implementation on Solid maintains compatibility with existing protocols, using the delegated-decentralised model, which provides better scalability as verified by our empirical evaluations. We discussed several features of the proposed solution that will benefit different stakeholders while respecting user autonomy (R2) and providing user-centric trust (R3), along with variants for adaptation to different PDS systems. Furthermore, we evaluated our proposed architecture using realistic scenarios, synthetic dataset generation with differential privacy, and gig worker empowerment, demonstrating the wide applicability of our architecture to high-impact collective privacy-preserving computation use cases. Through the case studies, we uncovered how real-life power dynamics can constitute the properties desirable by Libertas, and also how Libertas can achieve large-scale data utilisation while protecting both input and output privacy. The empirical results also verified scalability (R4), while shedding light on possible routes of optimisation when used in production.

Visionarily, our work offers a promising direction for empowering users with privacy-preserving and autonomy-respecting collaborative computation, while also incentivising the adoption of user-centric decentralised technologies like Solid, by demonstrating realistic use cases and unique benefits. This is crucial for large-scale, privacy-friendly, and mutually beneficial collaborative computation and data ecosystems, and could encourage data contribution with assured privacy and autonomy.

Limitations and Future Work

As our work provides a novel technical solution, we also note many interesting future research directions, as well as limitations in our current work. For example, decentralised systems may face complex governance challenges in ensuring accountability and responsibility , which is a very important direction of research. Further, while Libertas allows users to *express* and helps them to *utilise* their trusts, the current system does not provide additional support for users to *determine* and *maintain* them. It will be beneficial to explore alternative ways to express, manage and utilise trust of agents, such as different tiers of trust or dynamic preferences based on personnel, MPC computations and protocols, or "oathbreaker" recording for strengthened trust; users may also benefit from a policy checking step using a fine-grained policy language. Developing dedicated MPC protocols for such a context to utilise its trust dynamics will also be an interesting direction, which may also improve performance. It will also be useful to explore the performance in more broad contexts, such as in WAN settings and production settings; evaluating performance to other modalities of data (such as image data or streaming data) will also be interesting, as the current system focuses on non-real-time non-interactive collective data usage. Finally, appropriate user training, real-life collaboration and user studies will benefit the adoption and improvement of the architecture, as well as providing concrete proofs of the societal benefits provided by Libertas-like architectures.

Acknowledgments

This work is a part of the Ethical Web and Data Infrastructure in the Age of AI (EWADA) project, funded by Oxford Martin School. In addition, we express our special thanks to Prof. Malcolm Atkinson for his valuable suggestions in improving the presentation of this work.

Availability

The code and specifications of Libertas can be found at https://github.com/OxfordHCC/libertas. The evaluation logs will be made public in the published version of the paper, or upon request.

References

[1] 2023. Community Solid Server. https://github.com/CommunitySolidServer/CommunitySolidServer

[2] Aydin Abadi, Changyu Dong, Steven J. Murdoch, and Sotirios Terzis. 2022. Multi-party Updatable Delegated Private Set Intersection. In *Financial Cryptography and Data Security (Lecture Notes in Computer Science)*, Ittay Eyal and Juan Garay (Eds.). Springer International Publishing, Cham, 100–119. https://doi.org/10.1007/978-3-031-18283-9_6

[3] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.

[4] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. 2020. Secrets and likes: The drive for privacy and the difficulty of achieving it in the digital age. *Journal of Consumer Psychology* 30, 4 (2020), 736–758.

[5] Alessandro Acquisti, Curtis Taylor, and Liad Wagman. 2016. The Economics of Privacy. *Journal of Economic Literature* 54, 2 (June 2016), 442–492. https://doi.org/10.1257/jel.54.2.442

[6] Idris Adjerid, Eyal Peer, and Alessandro Acquisti. 2018. Beyond the Privacy Paradox. *MIS quarterly* 42, 2 (2018), 465–488.

[7] Hagar Afriat, Shira Dvir-Gvirsman, Keren Tsuriel, and Lidor Ivan. 2021. “This is capitalism. It is not illegal”: Users’ attitudes toward institutional privacy following the Cambridge Analytica scandal. *The Information Society* 37, 2 (2021), 115–127. https://doi.org/10/gn8nv3 Publisher: Routledge _eprint: https://doi.org/10.1080/01972243.2020.1870596.

[8] Wael Albayaydh and Ivan Flechais. 2023. Examining Power Dynamics and User Privacy in Smart Technology Use Among Jordanian Households. 4643–4659. https://www.usenix.org/conference/usenixsecurity23/presentation/albayaydh

[9] Nicolas Anciaux, Philippe Bonnet, Luc Bouganim, Benjamin Nguyen, Philippe Pucheral, Iulian Sandu Popa, and Guillaume Scerri. 2019. Personal Data Management Systems: The security and functionality standpoint. *Information Systems* 80 (Feb. 2019), 13–35. https://doi.org/10.1016/j.is.2018.09.002 TLDR: A general set of functionality and security requirements that any Personal Data Management System (PDMS) should consider is derived and a preliminary design for an extensive and secure PDMS reference architecture satisfying the considered requirements is proposed..

[10] Peter Baeck and Sophie Reynolds. 2020. Using collective intelligence to address climate change. https://www.themj.co.uk/Using-collective-intelligence-to-address-climate-change/218184

[11] Susan B. Barnes. 2006. A privacy paradox: Social networking in the United States. *First Monday* 11, 9 (Sept. 2006). https://doi.org/10.5210/fm.v11i9.1394 TLDR: The uproar over privacy issues in social networks is discussed by describing a privacy paradox; private versus public space; and, social networking privacy issues..

[12] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. 2006. Privacy and contextual integrity: framework and applications. In *2006 IEEE Symposium on Security and Privacy (S&P’06)*. IEEE, Berkeley/Oakland, CA, 15 pp.–198. https://doi.org/10.1109/SP.2006.32 TLDR: This work formalizes some aspects of contextual integrity in a logical framework for expressing and reasoning about norms of transmission of personal information to capture naturally many notions of privacy found in legislation, including those found in HIPAA, COPPA, and GLBA..

[13] Susanne Barth and Menno D. T. de Jong. 2017. The privacy paradox – Investigating discrepancies between expressed privacy concerns and actual online behavior – A systematic literature review. *Telematics and Informatics* 34, 7 (Nov. 2017), 1038–1058. https://doi.org/10.1016/j.tele.2017.04.013 TLDR: The overall findings of the systematic literature review will investigate the nature of decision-making (rational vs. irrational) and the context in which the privacy paradox takes place, with a special focus on mobile computing..

[14] James Bell, Adria Gascon, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Mariana Raykova, and Phillipp Schoppmann. 2022. Distributed, private, sparse histograms in the two-server model. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022).

[15] Tim Berners-Lee, Henry Story, and Sarven Capadisli. 2022. Web Access Control. https://solid.github.io/web-access-control-spec/

[16] Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. 2016. Label-Based Access Control: An ABAC Model with Enumerated Authorization Policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control (ABAC ’16)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/2875491.2875498 TLDR: This paper presents an ABAC model named LaBAC (Label-Based Access Control) which adopts the enumerated style for expressing authorization policies and shows equivalence of LaBac and 2-sorted-RBAC with respect to theoretical expressive power..

[17] G. R. Blakley. 1979. Safeguarding cryptographic keys. In *Managing requirements knowledge, international workshop on(AFIPS)*. IEEE Computer Society, 313. https://doi.org/10.1109/AFIPS.1979.98 Place: New York.

[18] Dan Bogdanov, Sven Laur, and Jan Willemson. 2008. Sharemind: A framework for fast privacy-preserving computations. In *European symposium on research in computer security*. 192–206. https://doi.org/10.1007/978-3-540-88313-5_13 tex.organization: Springer.

[19] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[20] Claire McKay Bowen and Joshua Snoke. 2019. Comparative study of differentially private synthetic data algorithms from the NIST PSCR differential privacy synthetic data challenge. *arXiv preprint arXiv:1911.12704* (2019).

[21] Lennart Braun, Daniel Demmler, Thomas Schneider, and Oleksandr Tkachenko. 2022. MOTION – A Framework for Mixed-Protocol Multi-Party Computation. *ACM Transactions on Privacy and Security* 25, 2 (2022), 8:1–8:35. https://doi.org/10.1145/3490390

[22] Dan Calacci. 2022. Organizing in the End of Employment: Information Sharing, Data Stewardship, and Digital Workerism. In *Proceedings of the 1st Annual Meeting of the Symposium on Human-Computer Interaction for Work (CHIWORK ’22)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3533406.3533424 TLDR: It is argued that worker-led technology design and data-driven research is a key step to ensure fair working futures under these conditions and for researchers in the CHI and CSCW fields to engage in a new kind of ”Digital Workerism”..

[23] Dan Calacci and Alex Pentland. 2022. Bargaining with the Black-Box: Designing and Deploying Worker-Centric Tools to Audit Algorithmic Management. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 1–24. https://doi.org/10.1145/3570601

[24] Dan Calacci and Jake Stein. 2023. From access to understanding: Collective data governance for workers. *European Labour Law Journal* 14, 2 (June 2023), 253–282. https://doi.org/10.1177/20319525231167981 Publisher: SAGE Publications.

[25] P Calcraft, I Thomas, M Maglicic, and A Sutherland. [n. d.]. Accelerating public policy research with synthetic data. https://www.adruk.org/fileadmin/uploads/adruk/Documents/Accelerating_public_policy_research_with_synthetic_data_December_2021.pdf. Accessed: 2024-12-10.

[26] Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, and Kjetil Kjernsmo. 2021. Solid Protocol Version 0.9.0, 2021-12-17. https://solidproject.org/TR/protocol

[27] Aaron Coburn, elf Pavlik, and Dmitri Zagidulin. 2022. Solid-OIDC. https://solidproject.org/TR/oidc.

[28] Cynthia L. Corritore, Beverly Kracher, and Susan Wiedenbeck. 2003. On-line trust: concepts, evolving themes, a model. *International Journal of Human-Computer Studies* 58, 6 (June 2003), 737–758. https://doi.org/10.1016/S1071-5819(03)00041-7 TLDR: The definitions of trust are analyzed, the relevant dimensions of trust for an on-line context are identified, and a definition of trust between people and informational or transactional websites is presented..

[29] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *Cryptology ePrint Archive* (2016).

[30] Natalia Criado and Jose M. Such. 2015. Implicit Contextual Integrity in Online Social Networks. *Information Sciences* 325 (Dec. 2015), 48–69. https://doi.org/10.1016/j.ins.2015.07.013 TLDR: The first computational model of Implicit Contextual Integrity is proposed, presenting an information model for Implicit contextual Integrity as well as a so-called Information Assistant Agent that uses the information model to learn implicit contexts, relationships and the information sharing norms in order to help users avoid inappropriate information exchanges and undesired information disseminations..

[31] Ivan Damgård, Kasper Damgård, Kurt Nielsen, Peter Sebastian Nordholt, and Tomas Toft. 2017. Confidential Benchmarking Based on Multiparty Computation. In *Financial Cryptography and Data Security (Lecture Notes in Computer Science)*, Jens Grossklags and Bart Preneel (Eds.). Springer, Berlin, Heidelberg, 169–187. https://doi.org/10.1007/978-3-662-54970-4_10

[32] Yves-Alexandre De Montjoye, Erez Shmueli, Samuel S Wang, and Alex Sandy Pentland. 2014. openpds: Protecting the privacy of metadata through safeanswers. *PloS one* 9, 7 (2014), e98790.

[33] Judith Wagner DeCew. 1997. *In Pursuit of Privacy: Law, Ethics, and the Rise of Technology.* Cornell University Press. Google-Books-ID: GjfkK56dtlAC.

[34] Tobias Dienlin and Miriam J. Metzger. 2016. An Extended Privacy Calculus Model for SNSs: Analyzing Self-Disclosure and Self-Withdrawal in a Representative U.S. Sample. *Journal of Computer-Mediated Communication* 21, 5 (Sept. 2016), 368–383. https://doi.org/10.1111/jcc4.12163 TLDR: A U.S. representative sample was used to test the privacy calculus' generalizability and extend its theoretical framework by including both self-withdrawal behaviors and privacy self-efficacy, and results confirmed the extended privacy calculus model..

[35] W. Diffie and M. Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory* 22, 6 (Nov. 1976), 644–654. https://doi.org/10.1109/TIT.1976.1055638 Conference Name: IEEE Transactions on Information Theory.

[36] Tamara Dinev and Paul Hart. 2006. An Extended Privacy Calculus Model for E-Commerce Transactions. *Information Systems Research* 17, 1 (March 2006), 61–80. https://doi.org/10.1287/isre.1060.0080 Publisher: INFORMS TLDR: Although Internet privacy concerns inhibit e-commerce transactions, the cumulative influence of Internet trust and personal Internet interest are important factors that can outweigh privacy risk perceptions in the decision to disclose personal information when an individual uses the Internet..

[37] Veena Dubal. 2023. On Algorithmic Wage Discrimination. *Columbia Law Review* 123, 7 (2023), 1929–1992. https://www.jstor.org/stable/27264954 Publisher: Columbia Law Review Association, Inc..

[38] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[39] Cynthia Dwork. 2008. Differential Privacy: A Survey of Results. In *Theory and Applications of Models of Computation (Lecture Notes in Computer Science)*, Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li (Eds.). Springer, Berlin, Heidelberg, 1–19. https://doi.org/10.1007/978-3-540-79228-4_1

[40] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography (Lecture Notes in Computer Science)*, Shai Halevi and Tal Rabin (Eds.). Springer, Berlin, Heidelberg, 265–284. https://doi.org/10.1007/11681878_14

[41] Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. 2020. *Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data.* OʹReilly, Beijing Boston Farnham Sebastopol Tokyo.

[42] Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. 2020. *Practical Synthetic Data Generation: Balancing Privacy and the Broad Availability of Data.* "O'Reilly Media, Inc.", Beijing Boston Farnham Sebastopol Tokyo.

[43] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. 2004. Efficient Private Matching and Set Intersection. In *Advances in Cryptology - EUROCRYPT 2004 (Lecture Notes in Computer Science)*, Christian Cachin and Jan L. Camenisch (Eds.). Springer, Berlin, Heidelberg, 1–19. https://doi.org/10.1007/978-3-540-24676-3_1

[44] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 169–178.

[45] Nina Gerber, Paul Gerber, and Melanie Volkamer. 2018. Explaining the privacy paradox: A systematic review of literature investigating privacy attitude and behavior. *Computers & Security* 77 (Aug. 2018), 226–261. https://doi.org/10.1016/j.cose.2018.04.002 TLDR: The privacy research community is suggested to agree on a shared definition of the different privacy constructs to allow for conclusions beyond individual samples and study designs, and provide strong evidence for the theoretical explanation approach called 'privacy calculus'..

[46] Srishti Gupta, Julia Jablonski, Chun-Hua Tsai, and John M. Carroll. 2022. Instagram of Rivers: Facilitating Distributed Collaboration in Hyperlocal Citizen Science. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW1 (2022), 97:1–97:22. https://doi.org/10.1145/3512944

[47] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. 2012. A Simple and Practical Algorithm for Differentially Private Data Release. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'12)*. Curran Associates Inc., Red Hook, NY, USA, 2339–2347.

[48] Benjamin Mako Hill and Aaron Shaw. 2021. The Hidden Costs of Requiring Accounts: Quasi-Experimental Evidence From Peer Production. *Communication Research* 48, 6 (Aug. 2021), 771–795. https://doi.org/10.1177/0093650220910345 Publisher: SAGE Publications Inc TLDR: It is concluded that requiring accounts introduces an undertheorized tradeoff for public goods production in interactive communication systems and

deters a large portion of low-quality participation..

[49] Jack Jamieson and Naomi Yamashita. 2023. Escaping the Walled Garden? User Perspectives of Control in Data Portability for Social Media. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 339:1–339:27. https://doi.org/10.1145/3610188 TLDR: This paper identifies current attitudes and practices toward controlling their social media data, and examines participants' impressions about the extent to which data portability may enhance their control, and proposes future directions for improving users' control in the context of social media..

[50] Caroline Jay, Robert Dunne, David Gelsthorpe, and Markel Vigo. 2016. To Sign Up, or not to Sign Up? Maximizing Citizen Science Contribution Rates through Optional Registration. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1827–1832. https://doi.org/10.1145/2858036.2858319 TLDR: It is demonstrated that removing the requirement to register increases the number of visitors to the site contributing to the project by 62%, without reducing data quality, and contribution rates are the same for people who choose to register, and those who remain anonymous..

[51] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. 2014. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584* (2014).

[52] Xin Jin, Ram Krishnan, and Ravi Sandhu. 2012. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. In *Data and Applications Security and Privacy XXVI (Lecture Notes in Computer Science)*, Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquin Garcia-Alfaro (Eds.). Springer, Berlin, Heidelberg, 41–55. https://doi.org/10.1007/978-3-642-31540-4_4 TLDR: This paper takes a step towards establishing formal connections between the three successful classical models and desired ABAC models by constructing an ABAC model that has "just sufficient" features to be "easily and naturally" configured to do DAC, MAC and RBAC..

[53] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. 2022. Synthetic Data – What, Why and How? https://doi.org/10.48550/arXiv.2205.03257 arXiv:2205.03257 [cs]

[54] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. 2022. Synthetic Data – what, why and how? https://doi.org/10.48550/arXiv.2205.03257 arXiv:2205.03257 [cs].

[55] Marcel Keller. 2020. MP-SPDZ: A Versatile Framework for Multi-Party Computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, Virtual Event USA, 1575–1590. https://doi.org/10.1145/3372297.3417872

[56] Marcel Keller, Emmanuela Orsini, and Peter Scholl. 2016. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 830–842. https://doi.org/10.1145/2976749.2978357

[57] Marcel Keller, Valerio Pastro, and Dragos Rotaru. 2018. Overdrive: Making SPDZ Great Again. In *Advances in Cryptology – EUROCRYPT 2018 (Lecture Notes in Computer Science)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 158–189. https://doi.org/10.1007/978-3-319-78372-7_6

[58] Sunnie S. Y. Kim, Elizabeth Anne Watkins, Olga Russakovsky, Ruth Fong, and Andrés Monroy-Hernández. 2023. Humans, AI, and Context: Understanding End-Users' Trust in a Real-World Computer Vision Application. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23)*. Association for Computing Machinery, New York, NY, USA, 77–88. https://doi.org/10.1145/3593013.3593978 TLDR: A holistic and nuanced understanding of trust in AI is provided through a qualitative case study of a real-world computer vision application, finding domain knowledge and context are important factors for trust-related assessment and decision-making..

[59] René F. Kizilcec. 2016. How Much Information? Effects of Transparency on Trust in an Algorithmic Interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2390–2395. https://doi.org/10.1145/2858036.2858402 TLDR: This work focuses on how transparent design of algorithmic interfaces can promote awareness and foster trust, using an online field experiment to test three levels of system transparency in the high-stakes context of peer assessment..

[60] Spyros Kokolakis. 2017. Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. *Computers & Security* 64 (Jan. 2017), 122–134. https://doi.org/10.1016/j.cose.2015.07.002 TLDR: The results of a review of research literature on the privacy paradox are presented and it is suggested that future studies should use evidence of actual behaviour rather than self-reported behaviour, and call for synthetic studies to be based on comprehensive theoretical models that take into account the diversity of personal information and the Diversity of privacy concerns..

[61] Konrad Kollnig, Reuben Binns, Max Van Kleek, Ulrik Lyngs, Jun Zhao, Claudine Tinsman, and Nigel Shadbolt. 2021. Before and after GDPR: tracking in mobile apps. *Internet Policy Review* 10, 4 (2021), 30.

[62] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).

[63] Nadin Kökciyan and Pinar Yolum. 2022. Taking Situation-Based Privacy Decisions: Privacy Assistants Working with Humans. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, Vienna, Austria, 703–709. https://doi.org/10.24963/ijcai.2022/99 TLDR: An agent-based model for a privacy assistant that can assess the trustworthiness of a situation even if the user has not interacted with the particular device before and can decide which situations are inherently ambiguous and thus can request the human to make the decision..

[64] Michelle S. Lam, Mitchell L. Gordon, Danaë Metaxa, Jeffrey T. Hancock, James A. Landay, and Michael S. Bernstein. 2022. End-User Audits: A System Empowering Communities to Lead Large-Scale Investigations of Harmful Algorithmic Behavior. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2 (Nov. 2022), 512:1–512:34. https://doi.org/10.1145/3555625

[65] Aliaksandr Lazouski, Fabio Martinelli, and Paolo Mori. 2010. Usage control in computer security: A survey. *Computer Science Review* 4, 2 (May 2010), 81–99. https://doi.org/10.1016/j.cosrev.2010.02.002

[66] Jie Li, Yongli Ren, and Ke Deng. 2022. FairGAN: GANs-based Fairness-aware Learning for Recommendations with Implicit Feedback. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. Association for Computing Machinery, New York, NY, USA, 297–307. https://doi.org/10.1145/3485447.3511958

[67] Yehuda Lindell. 2020. Secure multiparty computation. *Commun. ACM* 64, 1 (2020), 86–96. https://doi.org/10.1145/3387108

[68] Kaifeng Liu and Da Tao. 2022. The roles of trust, personalization, loss of privacy, and anthropomorphism in public acceptance of smart healthcare services. *Computers in Human Behavior* 127 (Feb. 2022), 107026. https://doi.org/10.1016/j.chb.2021.107026

[69] Miguel Malheiros and Sören Preibusch. 2013. Sign-Up or Give-Up: Exploring User Drop-Out in Web Service Registration. In *Symposium on Usable Privacy and Security (SOUPS)*. http://cups.cs.cmu.edu/soups/2013/trustbusters2013/Sign_up_or_Give_up_Malheiros.pdf

[70] Essam Mansour, Andrei Vlad Sambra, Sandro Hawke, Maged Zereba, Sarven Capadisli, Abdurrahman Ghanem, Ashraf Aboulnaga, and Tim Berners-Lee. 2016. A demonstration of the solid platform for social web applications. In *Proceedings of the 25th international conference companion on world wide web*. 223–226.

[71] Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An Integrative Model of Organizational Trust. *The Academy of Management Review* 20, 3 (1995), 709–734. https://doi.org/10.2307/258792 Publisher: Academy of Management.

[72] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. 94–103. https://doi.org/10.1109/FOCS.2007.66

[73] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. https://www.microsoft.com/en-us/research/publication/mechanism-design-via-differential-privacy/

[74] Yannic Meier and Nicole C. Krämer. 2024. The Privacy Calculus Revisited: An Empirical Investigation of Online Privacy Decisions on Between- and Within-Person Levels. *Communication Research* 51, 2 (March 2024), 178–202. https://doi.org/10.1177/00936502221102101 Publisher: SAGE Publications Inc TLDR: Results of a within-between random effects model showed that benefit perceptions were positively associated with self-disclosure intentions on between- and within-person levels and the privacy score was found to be effective in supporting users to make more privacy aware choices (within-person level)..

[75] Christian Meurisch, Bekir Bayrak, and Max Mühlhäuser. 2020. Privacy-preserving AI services through data decentralization. In *Proceedings of The Web Conference 2020*. 190–200.

[76] Christian Meurisch and Max Mühlhäuser. 2021. Data protection in AI services: a survey. *ACM Computing Surveys (CSUR)* 54, 2 (2021), 1–38.

[77] Payman Mohassel and Yupeng Zhang. 2017. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 19–38.

[78] Richard Mortier, Jianxin Zhao, Jon Crowcroft, Liang Wang, Qi Li, Hamed Haddadi, Yousef Amar, Andy Crabtree, James Colley, Tom Lodge, et al. 2016. Personal data management with the databox: What's inside the box?. In *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking*. 49–54.

[79] Andrew C. Myers and Barbara Liskov. 1997. A Decentralized Model for Information Flow Control. In *Proceedings of the Sixteenth ACM Symposium on Operating Systems Principles (SOSP '97)*. ACM, New York, NY, USA, 129–142. https://doi.org/10.1145/268998.266669 TLDR: This paper presents a new model for controlling information flo w in systems with mutual distrust and decentralized authority that improves on existing multilevel security models by allowing users to declassify information in a decentralized way, and by improving support for fine-grained data sharing..

[80] Helen Nissenbaum. 2004. Privacy as Contextual Integrity. *Washington Law Review* 79 (2004), 119. https://heinonline.org/HOL/Page?handle=hein.journals/washlr79&id=129&div=&collection=

[81] Helen Nissenbaum. 2011. A Contextual Approach to Privacy Online. *Daedalus* 140, 4 (Oct. 2011), 32–48. https://doi.org/10.1162/DAED_a_00113 TLDR: In developing this approach, the paper warns that the current bias in conceiving of the Net as a predominantly commercial enterprise seriously limits the privacy agenda, and proposes an alternative approach, rooted in the theory of contextual integrity..

[82] Goldreich Oded. 2009. Foundations of cryptography: volume 2, basic applications.

[83] Kieron O'Hara. 2016. The Seven Veils of Privacy. *IEEE Internet Computing* 20, 2 (March 2016), 86–91. https://doi.org/10.1109/MIC.2016.34 Conference Name: IEEE Internet Computing TLDR: This framework helps citizens think about when a privacy boundary is crossed or not, and why this differs not only across cultures, but also across generations and even for the same individuals..

[84] Open Data Institute and Projects by IF. 2022. *Perceptions of "Bottom-up Data Institutions": A report on research findings and key learnings for ODI.* Technical Report. Open Data Institute, London.

[85] Thomas F. J.-M. Pasquier, Jatinder Singh, David Eyers, and Jean Bacon. 2017. CamFlow: Managed Data-sharing for Cloud Services. *IEEE Transactions on Cloud Computing* 5, 3 (July 2017), 472–484. https://doi.org/10.1109/TCC.2015.2489211 arXiv: 1506.04391.

[86] Benny Pinkas. 2002. Cryptographic techniques for privacy-preserving data mining. *ACM Sigkdd Explorations Newsletter* 4, 2 (2002), 12–19.

[87] Michael O Rabin. 1981. *How to exchange secrets with oblivious transfer.* Technical Report. https://www.iacr.org/museum/rabin-obt/obtrans-eprint187.pdf

[88] Priscilla M. Regan. 1995. *Legislating Privacy: Technology, Social Values, and Public Policy.* University of North Carolina Press. https://www.jstor.org/stable/10.5149/9780807864050_regan

[89] Jens Riegelsberger, M. Angela Sasse, and John D. McCarthy. 2005. The mechanics of trust: A framework for research and design. *International Journal of Human-Computer Studies* 62, 3 (March 2005), 381–422. https://doi.org/10.1016/j.ijhcs.2005.01.001 TLDR: This work identifies contextual

properties and the actor's intrinsic properties that form the basis of trustworthy behavior and provides a frame of reference for the design of studies on trust in technology-mediated interactions, as well as a guide for identifying trust requirements in design processes..

[90] Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. 2018. Deepsecure: Scalable provably-secure deep learning. In *Proceedings of the 55th annual design automation conference*. 1–6.

[91] Rui Zhao, Malcolm Atkinson, Petros Papapanagiotou, Federica Magnoni, and Jacques Fleuriot. 2021. Dr.Aid: Supporting Data-governance Rule Compliance for Decentralized Collaboration in an Automated Way. In *The 24th ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*. https://doi.org/10.1145/3479604 TLDR: This work presents a framework, Dr.Aid, that helps individuals, organisations and federations comply with data rules, using automation to track which rules are applicable as data is passed between processes and as derived data is generated, arguing that this approach will lead to more agile, more productive and more trustworthy collaborations..

[92] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. 2016. Solid: a platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.* (2016).

[93] Shruti Sannon and Dan Cosley. 2022. Toward a More Inclusive Gig Economy: Risks and Opportunities for Workers with Disabilities. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 1–31. https://doi.org/10.1145/3555755

[94] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613. https://doi.org/10.1145/359168.359176 TLDR: This technique enables the construction of robust key management schemes for cryptographic systems that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces..

[95] Frank M. Shipman and Catherine C. Marshall. 2020. Ownership, Privacy, and Control in the Wake of Cambridge Analytica: The Relationship between Attitudes and Awareness. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376662 TLDR: Participants with the greatest awareness of the news story's details have more polarized attitudes about reuse, especially the reuse of content as data, and more willingness for social media platforms to demand corrections of inaccurate or deceptive content..

[96] H. Jeff Smith, Tamara Dinev, and Heng Xu. 2011. Information Privacy Research: An Interdisciplinary Review. *MIS Quarterly* 35, 4 (2011), 989–1015. https://doi.org/10.2307/41409970 Publisher: Management Information Systems Research Center, University of Minnesota TLDR: An interdisciplinary review of privacy-related research is provided in order to enable a more cohesive treatment and recommends that researchers be alert to an overarching macro model that is referred to as APCO (Antecedents ⟶ Privacy Concerns ⟶ Outcomes)..

[97] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*. 1451–1468.

[98] Jake M L Stein, Vidminas Vizgirda, Max Van Kleek, Reuben Binns, Jun Zhao, Rui Zhao, Naman Goel, George Chalhoub, Wael S Albayaydh, and Nigel Shadbolt. 2023. 'You are you and the app. There's nobody else.': Building Worker-Designed Data Institutions within Platform Hegemony. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–26. https://doi.org/10.1145/3544548.3581114 TLDR: A participatory design study in which platform workers deconstructed and re-imagined Uber's schema for driver data is conducted, and user-configurable tools for lightweight data institution building are proposed, as an alternative to redesigning existing platforms or delegating control to centralized trusts..

[99] Lauren Thornton, Bran Knowles, and Gordon Blair. 2022. The Alchemy of Trust: The Creative Act of Designing Trustworthy Socio-Technical Systems. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 1387–1398. https://doi.org/10.1145/3531146.3533196 TLDR: The concept of alchemy is introduced as the most apt characterization of a successful design process, illustrating the need for designers to engage with the richness of the trust landscape and creatively experiment with components from multiple models to create the perfect blend for their context..

[100] Carmela Troncoso, Mathias Payer, Jean-Pierre Hubaux, Marcel Salathé, James Larus, Edouard Bugnion, Wouter Lueks, Theresa Stadler, Apostolos Pyrgelis, Daniele Antonioli, Ludovic Barman, Sylvain Chatel, Kenneth Paterson, Srdjan Čapkun, David Basin, Jan Beutel, Dennis Jackson, Marc Roeschlin, Patrick Leu, Bart Preneel, Nigel Smart, Aysajan Abidin, Seda Gürses, Michael Veale, Cas Cremers, Michael Backes, Nils Ole Tippenhauer, Reuben Binns, Ciro Cattuto, Alain Barrat, Dario Fiore, Manuel Barbosa, Rui Oliveira, and José Pereira. 2020. Decentralized Privacy-Preserving Proximity Tracing. https://doi.org/10.48550/arXiv.2005.12273 arXiv:2005.12273 [cs].

[101] Virginia Balseiro, Timea Turdean, and Jeff Zucker. 2022. Solid WebID Profile. https://solid.github.io/webid-profile/

[102] Lijun Wei, Yuhan Yang, Jing Wu, Chengnian Long, and Bo Li. 2022. Trust Management for Internet of Things: A Comprehensive Study. *IEEE Internet of Things Journal* 9, 10 (May 2022), 7664–7679. https://doi.org/10.1109/JIOT.2021.3139989 Conference Name: IEEE Internet of Things Journal TLDR: The trust issue and the composition of trust management in IoT is comprehended, and the difference of existing work is illustrated, thereby motivating further research interest in this field..

[103] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[104] Andrew C. Yao. 1982. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. 160–164. https://doi.org/10.1109/SFCS.1982.38 ISSN: 0272-5428.

[105] Andrew Chi-Chih Yao. 1986. How to generate and exchange secrets. In *Foundations of computer science, 1986., 27th annual symposium on*. 162–167. tex.organization: IEEE.

[106] Zheng Yao, Silas Weden, Lea Emerlyn, Haiyi Zhu, and Robert E Kraut. 2021. Together But Alone: Atomization and Peer Support among Gig Workers. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–29. Publisher: ACM New York, NY, USA.

[107] Angie Zhang, Alexander Boltz, Chun Wei Wang, and Min Kyung Lee. 2022. Algorithmic Management Reimagined For Workers and By Workers: Centering Worker Well-Being in Gig Work. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–20. https://doi.org/10.1145/3491102.3501866

[108] Jun Zhao, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. 2016. Privacy Languages: Are we there yet to enable user controls?. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 799–806. https://doi.org/10/gmbqwn

[109] Rui Zhao and Jun Zhao. 2024. Perennial Semantic Data Terms of Use for Decentralized Web. In *Proceedings of The ACM Web Conference 2024*. ACM, Singapore, 2238 – 2249. https://doi.org/10.1145/3589334.3645631

[110] Yuchen Zhao, Hamed Haddadi, Severin Skillman, Shirin Enshaeifar, and Payam Barnaghi. 2020. Privacy-preserving activity and health monitoring on databox. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20)*. ACM, Heraklion Greece, 49–54. https://doi.org/10.1145/3378679.3394529

[111] Serena Zheng, Noah Apthorpe, Marshini Chetty, and Nick Feamster. 2018. User Perceptions of Smart Home IoT Privacy. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 200:1–200:20. https://doi.org/10.1145/3274469 arXiv: 1802.08182 TLDR: This study conducts eleven semi-structured interviews with smart home owners, investigating their reasons for purchasing IoT devices, perceptions of smart home privacy risks, and actions taken to protect their privacy from those external to the home who create, manage, track, or regulate IoT devices and/or their data..

[112] Valentin Zieglmeier and Alexander Pretschner. 2023. Rethinking People Analytics With Inverse Transparency by Design. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (2023), 292:1–292:29. https://doi.org/10.1145/3610083

[113] Shoshana Zuboff. 2019. *The age of surveillance capitalism: The fight for a human future at the new frontier of power: Barack Obama's books of 2019*. Profile books.

MWEM Algorithm

The Multiplicative Weights and Exponential Mechanism (MWEM) algorithm is an iterative algorithm designed to generate a synthetic dataset whose responses to queries closely resemble those of the original dataset [47]. MWEM operates by taking the original dataset as input D and a set Q of linear queries. A linear query, also known as a counting or statistical query, is defined by a function q that maps data records to the interval $[-1, 1]$. The answer to a linear query on a dataset D, denoted by $q(D)$, is the sum $\sum_{x \in D} q(x) \cdot D(x)$. Additionally, two parameters of interest in the algorithm are the epsilon value ϵ (privacy parameter) and the number of iterations T. The algorithm produces a distribution A over $\mathcal{D}$ such that the difference between $q(A)$ and $q(D)$ is small. MWEM repeatedly samples a query for which the difference is still large and updates the weight that A places on each record x. MWEM satisfies ϵ-differential privacy by leveraging the exponential mechanism [73] to sample queries and the Laplace mechanism [40] to add noise to the query results. The pseudocode of the algorithm is as follows. We implemented this as an MPC circuit without special treatments such as optimisation.

Algorithm 1: The MWEM algorithm [47]

```
Input  : Dataset $D$ over a universe $\mathcal{D}$,
         Set of linear queries $Q$,
         Number of iterations $T$,
         Privacy parameter $\epsilon > 0$.
Let $n$ denote $|D|$, the number of records in $D$.
Let $A_0$ denote $n$ times the uniform distribution over $\mathcal{D}$.
for $i \in \{1, \ldots, T\}$ do
    Exponential Mechanism: sample a query $q_i \in Q$ using the Exponential Mechanism parametrized with
      epsilon value $\epsilon/2T$ and the score function: $s_i(D, q) = |q(A_{i-1}) - q(D)|$
    Laplace Mechanism: Let measurement $m_i = q_i(D) + \mathsf{Lap}(2T/\epsilon)$
    Multiplicative Weights: Let $A_i$ be $n$ times the distribution whose entries satisfy
        $$A_i(x) \propto A_{i-1}(x) \times \exp(q_i(x) \times (m_i - q_i(A_{i-1}))/2n)$$
end
Output : $A = A_T$ (or $A = \text{avg}_{i<T} A_i$)
```