Data Protection Intents for Software-Defined Networking

Benjamin E. Ujcich and William H. Sanders
Information Trust Institute and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois USA
E-mail: {ujcich2, whs}@illinois.edu

***Abstract*—The rise of intent-based networking (IBN) allows enterprises to use software-defined networking (SDN) architectures to specify what network requirements are needed rather than specify how such requirements will be implemented. For enterprises that process personal data, those network requirements must necessarily consider data protection by design to comply with new regulations such as the European Union's GDPR.**

We argue that the centralized data plane view of SDN architectures and the network intent abstractions of IBN can aid in the design of systems that require data protection. We propose a data protection intent framework that leverages SDN and network intents. We use the GDPR as a representative data protection framework and identify the applicable regulatory requirements for system and network design. Based on those requirements, we design an SDN-based architecture for data protection intents that allows data services to request network resources by using data protection abstractions. We implement a proof-of-concept network application for the ONOS SDN controller and explain how our framework can be useful in a representative data breach case study to aid in responding to regulator requests.

***Index Terms*—network intent, software-defined networking, SDN, intent-based networking, IBN, data protection, GDPR, regulatory compliance, logging**

I. Introduction

Software-defined networking (SDN) provides flexible support for programmatic control over enterprise networks. One of the key benefits of the SDN architecture, as compared to traditional enterprise network architectures, is the centralization of decision-making into a logically centralized *SDN controller* that learns information about data plane activities and implements forwarding rules within network devices to enforce desired data plane configurations. Controllers provide abstractions such that developers and practitioners can specify configurations through high-level *network intents* without the need to understand the low-level mechanisms of how such intents are implemented on network devices [1].

While such abstractions simplify network design and operations for practitioners, challenges remain for enterprises that manage personal data. In particular, the level of abstraction at which SDN controllers perform their activities may not be the right one to support meaningful reasoning about such activities as related to personal data. For instance, suppose that an enterprise network's security practitioner discovers a security breach that was enabled by and affects data plane hosts. The low-level network configuration details (e.g., flow rules installed at the time of the breach) provide a snapshot of the network's state at the time of the breach but may not effectively describe what data were traversing the network or why such traversal was allowed at a human-understandable level of abstraction (e.g., a compromised end host exfiltrated personal data from a database server).

For enterprises that must handle personal data, the ability to provide information about such high-level network activities accurately and quickly[1] is not only beneficial for assessing situational awareness and risk, but, increasingly required by regulation. Recent data protection regulations, most notably the European Union's General Data Protection Regulation (GDPR) [4], require that enterprises be able to answer questions about the extent to which data are stored, processed, and transferred in order to demonstrate compliance through accountability; non-compliance may result in extensive monetary penalties and reputational damage. Thus, enterprises that want to ensure compliance must consider systems and networks that are cognizant of data protection requirements *by design*.

To support that goal, we believe that SDN is well-positioned in the design of such systems that must consider data protections. First, the network's role in "seeing" all data plane activities and communications [5] allows for greater assurances about completeness in recording what network activities have occurred, because all flow rules originate from a logically centralized SDN controller. A data-protection-aware system can leverage the SDN controller to interpose on all end host communication that involves personal data, and that simplifies the record-keeping needed to demonstrate compliance. Second, network softwarization can bridge the abstractions between low-level network state details and high-level application details as related to personal data. Systems can intelligently incorporate the network when requesting the network's resources. Network intents can describe what data flow and the purposes for which those data are being requested.

In this paper, we propose a *data protection intent* framework and system architecture for SDN. Using the GDPR as a motivating example of a comprehensive data protection regulatory

[1]Recent large-scale data breaches, such as the Starwood Hotels data breach reported in November 2018 that affected up to 500 million guests' personal data, involved attacks that encrypted personal data prior to exfiltration [2]. As a result, the exfiltration avoided detection for approximately four years [3].

TABLE I
SUMMARY OF GDPR NETWORK REQUIREMENTS APPLICABLE TO DATA PROCESSING AND DATA TRANSFERS.

Design Component	Requirements
Data usage	• Lawfulness of processing [4, Art. 6]
Logging, analysis, and forensics	• Demonstration of accountability [4, Art. 5] • Recording of data processing [4, Art. 30] • Processing responsibilities [4, Rec. 82] • Notification of data breaches [4, Art. 33] • Monitoring of compliance by data protection officer [4, Arts. 37–39]
Access control	• Design of data protection [4, Art. 25] • Technical measures [4, Rec. 78]

framework, we show how the GDPR's requirements are applicable to networks and their design. In addition to traditional security requirements such as access control, we find that SDN can efficiently track data transfers because of the controller's centralized view and because of its programmable nature, which bridges levels of abstraction as necessary to reason about regulatory compliance. Our contributions include 1) an overview of networks' role in the GDPR (Section II-A); 2) a framework and system architecture design that incorporates data protection intents (Section III); 3) a proof-of-concept data protection intent app for the ONOS SDN controller [6] that interposes on all data protection intent requests (Section IV); and 4) a representative data breach case study to demonstrate the utility of our framework (Section V).

II. BACKGROUND AND RELATED WORK

A. Data Protection and the GDPR

Data protection encompasses the mechanisms by which personal data are protected against unlawful use, processing, or transfer to ensure the rights of *data subjects* whom the data are about and *data controllers* that process such data. Regulations for data protection can be domain-specific (e.g., the U.S. Health Insurance Portability and Accountability Act, or HIPAA, for personal health records) or general (e.g., the EU-based General Data Protection Regulation, or GDPR).

A study by the European Union Agency for Network and Information Security found that "data protection features are, on the whole, ignored by traditional engineering approaches when implementing the desired functionality" [7], which suggests that a gap remains between the legal and technical domains. SDN can be used to comply with HIPAA to enforce data plane isolation [8], [9] and secure transmission [10]. Design requirements for SDN accountability have been proposed within a general accountability framework designed to support compliance with regulatory standards [11].

We focus on the GDPR[2] and summarize the relevant system and network design requirements for data processing (with particular focus on data transfers) in Table I. The GDPR broadly defines *processing* as an "operation . . . performed on personal data . . . such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, [and] dissemination or otherwise making available" [4, Art. 4]. Thus, processing affects not only the storage of personal data (e.g., databases) but also the transfer of personal data (e.g., traversal of data over networks).

With the GDPR, data usage must be justified in order for data to be processed lawfully, such as through a subject's consent or a controller's legal obligation [4, Art. 6]. Given the centralized logical view of the network available with SDN, the lawfulness of data processing can be recorded and evaluated at the time that personal data traverse the network. As we show later in Sections III and IV, an SDN network application can maintain a set of references and pointers among personal data, lawful bases for processing, and records of which network mechanisms were implemented (i.e., network intents and flow rules) to demonstrate accountability [4, Art. 5].

Data controllers must keep records of processing [4, Art. 30] to demonstrate accountability to regulators [4, Art. 5], to monitor for compliance within controller organizations by data protection officers [4, Arts. 37–39], and to understand the extent to which data are affected when breaches occur [4, Art. 33]. While traditional networking architectures may be able to record some network events at various levels of abstraction (e.g., NetFlow records), SDN's programmability is well-suited not only for recording low-level network events but also for bridging the semantic gap with high-level data-processing events to produce richer logs and records. Such semantic understanding benefits root-cause analysis and forensics.

Enforcement of access control of personal data can be ensured through proper network isolation [4, Art. 25]. SDN-based network isolation [8], [9], SDN-based network access control [14], [15], and application access control using SDN [16] have been studied extensively in prior work; we do not focus here on novel access control designs.

B. Intent-Based Networking

Intent-based networking (IBN) abstracts the mechanisms of *how* networks are configured to focus on the specification of high-level goals of *what* the network ought to achieve. Two major open-source SDN controllers, ONOS [6] and OpenDaylight [17], implement their own intent frameworks. ONOS's Intent Framework uses intents to describe network resources, constraints, matching criteria, and treatments to be applied to traffic [18]. OpenDaylight's Network Intent Composition provides similar functionality, particularly for network orchestration and business applications [19].

IBN has been proposed in use cases such as service chaining, end host security monitoring, and network isolation [20]. To the best of our knowledge, we are the first to extend an intent framework specifically for data protection.

III. DATA PROTECTION INTENT FRAMEWORK FOR SDN

We now present our framework for data protection intents that use SDN. In our design, we consider four goals, with

[2]Similar general data protection legislation have been proposed elsewhere, such as the Personal Data Protection Bill 2018 in India [12] and the California Consumer Privacy Act of 2018 in the U.S. state of California [13]. We limit our analysis to the GDPR for scoping reasons but note that the requirements we analyze may be applicable to other general data protection frameworks.

TABLE II
DATA PROTECTION INTENT FIELDS.

Field	Description
intentID	Data protection intent unique identifier
sourceServiceID	Data service source identifier
destinationServiceID	Data service destination identifier
dataIDs	Pointers to identifiers of personal data
lawfulBasisIDs	Pointers to identifiers of lawful bases that allow data to be processed or transferred
purposeIDs	Pointers to identifiers of specific purposes for which data are to be processed or transferred
intentObjects	Associated network intent objects that implement data plane network connectivity

practitioners and regulatory compliance in mind:

G1 Specify host-to-host data plane connection requests for data processing through data protection abstractions.

G2 Maintain observance and oversight over all network requests that involve personal data.

G3 Log control plane and data plane events to bridge network abstractions and data protection abstractions.

G4 Use data protection abstractions to query past events for analysis and forensics.

A. Data Model

Prior work [21]–[24] models GDPR concepts in data processing workflows. Such models provide a way to automatically describe, reason about, and query data protection metadata. Given the use of networks to transfer data and the SDN role of "seeing" all data plane activities, we have adapted the relevant portions of these data models as they relate to data transfers. We created a data structure class, DataProtectionIntent, that encompasses the necessary data protection abstractions. Table II lists the class fields. We assume that an enterprise uses unique identifiers to reference personal data objects, the lawful bases for determining which data are allowed to be processed or transferred (e.g., a reference to a subject's consent), and the purposes for which data are allowed to be processed or transferred (e.g., disclosure to a third party).

B. System Architecture

Figure 1 shows an overview of the system design of the data protection intent framework. Figure 2 shows the abstraction layers across different contexts. Next, we explain the roles of the system components.

1) Data Services: Data services are client and server processes on end hosts that use, process, store, and transform personal data as defined in the GDPR. For instance, a Web application that collects personal data and a back-end database that stores personal data are both data services. Data services submit their data protection intents to the data protection intent app to request network connections to other data services. Data services need not understand the underlying network mechanisms or topology, as these details are abstracted. Thus, data services can express their data processing and transfer intents through data protection abstractions (goal **G1**).

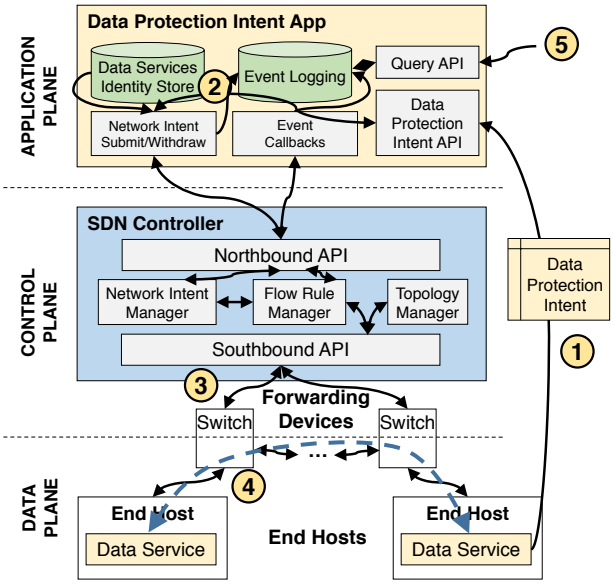

Fig. 1. Data protection intent system architecture. **1:** A data service (client) submits a data protection intent request to the data protection intent app. **2:** The data protection intent app processes the request. **3:** The SDN controller compiles and installs network intents in the data plane's devices through flow rules. **4:** The data services (client and server) transmit data through the data plane paths created by network intents (represented by the dashed arrow). **5:** A practitioner can query the data protection intent app for analysis or to fulfill regulator requests for records.

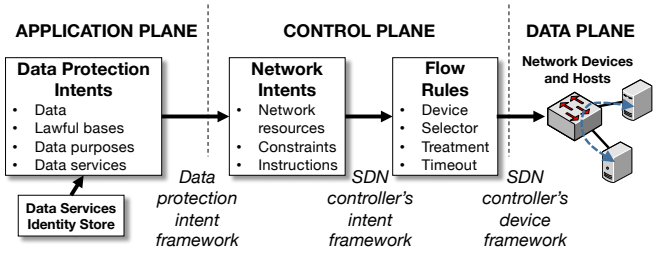

Fig. 2. Abstraction layers for data protection that uses intent-based SDN. Data protection intents are concerned with data processing and transfers among data services; network intents are concerned with end-to-end network connectivity; and flow rules are concerned with data plane processing in network devices.

2) Data Protection Intent Application: The data protection intent app is an SDN network application that receives data protection intent requests and communicates with the SDN controller's northbound API interface to implement such requests in the network. The data protection intent app

1) interposes on all relevant personal data transfers in the system to ensure completeness of oversight (goal **G2**);
2) maintains a data services identity store that maps data service identifiers with their network identifiers (e.g., a data service's host name, IP address, and port); and
3) logs all data protection intent requests and network state changes (goal **G3**) to be used for later queries (goal **G4**).

Because the data protection intent app concerns itself only

with end-to-end path connectivity through network intents, the data protection intent app need not compute such a path itself; instead, the data protection intent app uses the controller's intent framework to build the resulting data plane path.

Data services can request that the data protection intent app update previously submitted data protection intents. For instance, if a database server and Web client exchange additional data over an existing connection, then the data protection intent can be updated to represent the additional data exchange; there is no need to install additional network intents.

3) SDN Controller: The SDN controller coordinates the northbound and southbound APIs. The controller exposes its intent framework's service methods as API calls (e.g., submit() and withdraw() network intents) and allows apps to register callback methods whenever an intent is updated (e.g., a topology change). The data protection intent app registers event callbacks to monitor network changes.

IV. Implementation

We used the ONOS SDN controller's application and network intent frameworks to design a proof-of-concept implementation of a data protection intent app. ONOS network applications can be installed in the controller as event-driven modular components that subscribe to various event listeners.

The app's main class, DataProtectionIntentManager, coordinates the data protection intent service. The class's primary public methods, which are submitDPI(), withdrawDPI(), updateDPI(), and queryDPI(), take care of submitting, withdrawing, updating, and querying of DataProtectionIntent objects, respectively. Listener methods implement callbacks to record network intent and flow rule events.

For each submitted DataProtectionIntent object, the app maintains a list of associated (network) Intent objects. As the network's state may change over time because of link, device, or host failures, the network intents service may need to reconfigure flow rules to continue to provide connectivity. The app uses its IntentListener listener to check whether any network intent changes affect corresponding data protection intents; if they do, the app records the state changes.

We assume that not all data services will provide implementations for generating data protection intents. To overcome that limitation, the data protection intent app subscribes to flow rule events by using its FlowRuleListener listener. The listener filters flow rule events that match data service identities from the data services identity store. Thus, the app meets goal **G2** of ensuring the completeness of record-keeping, even if the abstraction mappings among data protection, network intents, and flow rules cannot be determined automatically.

Data services can update their data protection intents by calling updateDPI() with the data protection intent identifier and any data, lawful basis, or purpose pointers to be added.

We imagine that it will be desirable for scaled implementations to consider end-to-end latency as a performance metric, measured as the time from when one data service requests an intent to the time when data begin to flow between services. The extent to which that overhead can be reduced will vary, but will be critical to avoiding the interruption of normal network operations.

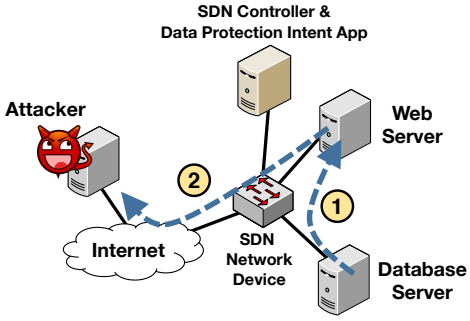

Fig. 3. Data flows showing an attacker's exfiltration of personal data. Solid lines represent logical network connections, and dashed arrows represent flows created through data protection intents. **1:** A data protection intent requests a data transfer from the database to the Web server. **2:** A data protection intent requests a data transfer from the Web server to an external entity.

V. Case Study: Data Breach and Data Exfiltration

We consider a representative case study of how our data protection intent framework would be used during a suspected data breach. We present a scenario that affects personal data within an enterprise and analyze how our framework can be used by practitioners, data protection officers, and regulators.

A. Scenario

Suppose that an enterprise, acting as a data controller, maintains personal data about data subjects for its business use. The personal data are stored in a back-end database server, which functions as a data service that issues data protection intents for data requests. Another data service, a front-end Web server that receives, processes, and transmits data to the database server, also issues data protection intents. Data subjects can send requests to the Web server to update or view personal data that are stored in the database.

An attacker wishes to exfiltrate personal data about many data subjects from the enterprise. The attacker uses a priori stolen credentials of a data subject to log in to the Web server's website and exploit vulnerabilities in the Web and database servers (e.g., SQL injection) to gain further access and exfiltrate personal data. Figure 3 shows the relevant data transfers. The attacker's actions cause personal data to be sent from the database server to the Web server, and the Web server returns the personal data to the external attacker.

B. Analysis

Without the data protection intent framework, the security practitioner's task of determining the data breach's extent would be complicated. The practitioner would have to reconstruct relevant events manually by inferring causal relations among different records from host, server, and network logs.

With the data protection intent framework (implemented across data services) and the data protection intent app, the practitioner can query the data protection intent app to find any relevant data protection intents that may have been submitted.

Next, the practitioner can link the data protection intent to the data that were exfiltrated and to the history of any network state changes (e.g., network intent state changes, flow rule changes) that were performed to allow the exfiltration. That allows the practitioner to rule out which data were *not* exfiltrated and thus which data and data services were not at risk. The GDPR mandates that known data breaches be reported to regulators within 72 hours of discovery [4, Art. 33]. Our framework allows the practitioner to assess the extent of damage more easily because he or she can link personal data to network state changes to understand the event semantics more fully. Finally, the practitioner provides the records to the enterprise's data protection officer so that the data controller can demonstrate accountability to regulators.

VI. Discussion and Future Work

A limitation of our framework is that its implementation alone will not satisfy all of the GDPR's requirements. Certain GDPR requirements involve organizational management practices rather than technical mechanisms to ensure compliance [21], [24], though we believe that our framework can mitigate some of the difficulties in compliance verification by bridging semantic gaps in record-keeping.

Our system does not consider data transfer and processing events that do not use the network. We envision that distributed GDPR-aware middleware tools will be necessary so that the data services can tag personal data with descriptions of the services' processing within data processing workflows. SDN can fill an important role if the SDN controller is allowed to serve as an online reference monitor for real-time access control enforcement of regulatory compliance policies.

We assume that data services report correct information in their data protection intents and that the data protection intent app infrastructure itself is adequately secured from attacks. Our future work will consider adversarial threat models and designs in which we reduce the size of the system's trusted computing base (TCB). Prior work has shown that certain integrity guarantees can be made with respect to storage of data controller records for the GDPR with distributed consensus ledgers [25].

VII. Conclusion

Data protection continues to be a highly relevant topic for enterprises that maintain personal data, especially as such data are increasingly subject to data protection regulations such as the GDPR. Given the complexities involved in understanding how an enterprise's network events correspond to processing and transfer of personal data within data-processing workflows, we proposed and implemented a data protection intent framework to simplify that task by leveraging the centralized control and oversight of SDN and IBN.

Acknowledgment

The authors would like to thank Jenny Applequist for her editorial assistance.

References

[1] D. Sanvito, D. Moro, M. Gull, I. Filippini, A. Capone, and A. Campanella, "ONOS intent monitor and reroute service: Enabling plug and play routing logic," in *Proceedings of NetSoft '18*. IEEE, June 2018, pp. 272–276.

[2] B. Krebs, "Marriott: Data on 500 million guests stolen in 4-year breach." [Online]. Available: https://krebsonsecurity.com/2018/11/marriott-data-on-500-million-guests-stolen-in-4-year-breach/

[3] T. Johnson, "Hackers lurked undetected on networks now owned by Marriott for 4 years," McClatchy DC Bureau, Nov. 2018.

[4] Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016," in *Official Journal of the European Union*, vol. L 119, May 2016, pp. 1–88.

[5] A. Bates, K. Butler, A. Haeberlen, M. Sherr, and W. Zhou, "Let SDN be your eyes: Secure forensics in data center networks," in *Proceedings of USENIX SENT '14*. USENIX Association, Apr. 2014.

[6] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed SDN OS," in *Proceedings of HotSDN '14*. ACM, 2014, pp. 1–6.

[7] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, and S. Schiffner, "Privacy and data protection by design — from policy to engineering," European Union Agency for Network and Information Security, Tech. Rep., Dec. 2014.

[8] R. Soulé, S. Basu, P. J. Marandi, F. Pedone, R. Kleinberg, E. G. Sirer, and N. Foster, "Merlin: A language for provisioning network resources," in *Proceedings of CoNEXT '14*. ACM, 2014, pp. 213–226.

[9] S. Gutz, A. Story, C. Schlesinger, and N. Foster, "Splendid isolation: A slice abstraction for software-defined networks," in *Proceedings of HotSDN '12*. ACM, 2012, pp. 79–84.

[10] P. Li, C. Xu, Y. Luo, Y. Cao, J. Mathew, and Y. Ma, "CareNet: Building a secure software-defined infrastructure for home-based healthcare," in *Proceedings of SDN-NFVSec '17*. ACM, 2017, pp. 69–72.

[11] B. E. Ujcich, A. Miller, A. Bates, and W. H. Sanders, "Towards an accountable software-defined networking architecture," in *Proceedings of NetSoft '17*. IEEE, July 2017, pp. 1–5.

[12] Parliament of India, "The Personal Data Protection Bill, 2018," 2018.

[13] California State Legislature, "Assembly Bill No. 375 (California Consumer Privacy Act of 2018)," Jun. 2018.

[14] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "FRESCO: Modular composable security services for software-defined networks," in *Proceedings of NDSS '13*. Internet Society, 2013.

[15] H. Hu, W. Han, G.-J. Ahn, and Z. Zhao, "FlowGuard: Building robust firewalls for software-defined networks," in *Proceedings of HotSDN '14*. ACM, 2014, pp. 97–102.

[16] T. O'Connor, W. Enck, W. M. Petullo, and A. Verma, "PivotWall: SDN-based information flow control," in *Proceedings of SOSR '18*. ACM, 2018, pp. 3:1–3:14.

[17] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proceedings of WoWMoM '14*. IEEE, June 2014, pp. 1–6.

[18] ONF, "ONOS intent framework." [Online]. Available: https://wiki.onosproject.org/display/ONOS/Intent+Framework

[19] OpenDaylight Project, "Network Intent Composition: Main." [Online]. Available: https://wiki.opendaylight.org/view/Network_Intent_Composition:Main

[20] ——, "Network Intent Composition: Use Cases." [Online]. Available: https://wiki.opendaylight.org/view/Network_Intent_Composition:Use_Cases

[21] B. E. Ujcich, A. Bates, and W. H. Sanders, "A provenance model for the European Union General Data Protection Regulation," in *Proceedings of IPAW '18*. Springer, 2018, pp. 45–57.

[22] H. J. Pandit and D. Lewis, "Modelling provenance for GDPR compliance using linked open data vocabularies," in *Proceedings of Society, Privacy and the Semantic Web - Policy and Technology '17*, 2017.

[23] C. Bartolini, R. Muthuri, and C. Santos, "Using ontologies to model data protection requirements in workflows," in *Proceedings of New Frontiers in Artificial Intelligence*. Springer, 2017, pp. 233–248.

[24] D. Basin, S. Debois, and T. Hildebrandt, "On purpose and by necessity: Compliance under the GDPR," in *Proceedings of Financial Cryptography and Data Security '18*, Mar. 2018.

[25] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," in *Proceedings of ARES '17*. ACM, 2017, pp. 14:1–14:10.