The Ethical Black Box: Preserving Data Self-sovereignity via Access Control based on Blockchain*

Daniel Henselmann[1][0000−0001−6701−0287], Delfina Sol Pandiani[2][0000−0003−2392−6300], Laura Waltersdorfer[3][0000−0002−6932−5036], Lu-Chi Liu[4][0000−0003−0249−1491], Sindhu Vasireddy[5][0000−0002−3522−5504], and John Domingue[6][0000−0001−8439−0293]

[1] Fraunhofer Institute for Integrated Circuits IIS, Nürnberg, Germany
[2] University of Bologna, Bologna, Emilia-Romagna, Italy
[3] TU Wien, Vienna, Austria
[4] University of Amsterdam, Amsterdam, Netherlands
[5] University of Ghent, Ghent, Belgium
[6] Knowledge Media Institute, Open University, UK

Abstract. Personal user data is valuable and the main focus of many current business models. However, only service providers collecting data are currently in charge of managing and exploiting it since centralised data storage often prevents users from deciding on their private data in a self-sovereign way. Furthermore, the collection of empirical data on executions on personal data is cumbersome and seldom done. Thus, we investigate in this work how decentralised technologies (i.e. Blockchain, Solidity), ontologies and functional languages for executable permission controls (e.g.: eFlint for Smart Contracts) can enable the logging of empirical executions in a privacy-preserving manner. Our contributions are i) the eFlint Ontology, mapping eFlint specifications to RDF and ii) an architecture design combing the different technologies to show the feasibility of supporting self sovereignty in terms of personal data management.

Keywords: Solid · eFLINT · Blockchain · Smart Contract · Merkle Tree · Semantic Web · Ontology · Legal Consent

1 Research Questions

RQ1 *How can we support citizens to be self-sovereign in terms of their data?*
RQ2 *What are possible ways to regulate/control the empirical use of semantic data?*
RQ3 *How can specific decentralized technologies (Solid and blockchain) be used to achieve self-sovereignty in terms of user data and what are its limitations?*

* Technical report of the task force Dragon from ISWS 2022 led by John Domingue

2 Empirical Semantics

The majority of the approaches to empirical semantics investigate static semantic data snapshots (e.g., a RDF dataset). In contrast in this work, we are looking at the empirical semantics associated with *executions* of a semantic processor over data which is often private and thereby requires consent. Within this general area, our focus has been on setting up mechanisms which allow the empirical execution of semantics to be captured in a privacy-preserving verifiable manner and which gives citizens data self-sovereignty. From a technical point of view, we have investigated how ontologies, decentralised technologies, and a formal language can support the above. More specifically, Solid provides decentralised data storage and blockchain a decentralised trusted public record. Additionally, eFlint, a functional language provides a formal way to capture user legal consent of data use in a manner that can be compiled to run on the Blockchain as a Smart Contract. In summary, our approach allows us to empirically examine the traces and logs of semantic executions over private data, while respecting consent.

3 Introduction

Personal user data is extremely valuable, to the extent that the current business model landscape is increasingly dependent on personal data exploitation. This exploitation relies in great part on citizens' unawareness, ignorance of, or inability to understand the way that their data is produced and used. Critically, the empirical use of (collection, processing, etc.) citizens' personal data by parties (e.g., companies, organizations, governments) is neither easily traceable nor currently logged in an understandable way.

Part of the issue is that citizens are not really the owners of their, data due to the current landscape of data centralization. Sir Tim Berners-Lee, views the situation as a demonstration "that the Web had failed instead of served humanity, (...), and failed in many places" [4]. The failure corresponds in part to the fact that the, oftentimes non-consensual, exploitation of personal data is not only performed for marketing purposes, but also for more far-reaching and sensitive issues such as political elections and health care and reproductive justices (e.g., Cambridge Analytica scandal, post-Roe v. Wade criminal charges [12, 9]).

An alternative to data centralization in the hands of powerful parties would be to support citizens to be self-sovereign owners and controllers of their personal data, allowing them to hold companies and organizations using such personal data accountable. With the emergence of GDPR and similar initiatives, efforts have been made to mitigate the effects of non transparent personal data exploitation. However, we are still lacking technical infrastructure that allows for an effective, trustworthy, and ethical way of managing and logging empirical personal data use, in compliance with fine-granular consents of citizens. We see decentralized technologies as critical tools to create an ethical privacy preserving black box for data and semantic use. Specifically, we aim to investigate how

the storing of semantic data processing compliance on the blockchain enables semantics based on observation (aka empirical semantics).

Traditional access control systems are lacking the capabilities we need to achieve self sovereign data management by users. Distributed technologies, such as Blockchain-based approaches, Semantic Web and Solid are already providing solutions for modern access control systems, while providing features such as immutability, logging of records analytical tasks and execution of agreed obligations. Our used methods and data include:

1. **eFLINT** is a domain-specific language for formalizing executable norm specifications like GDPR with support for reasoning [2].
2. A **Merkle Tree** is a tree structure of hash values where the leaves contain the hash value of a data block and all nodes further up the tree contain hash values built from the hashes in its children up to the root, [15], which can be used to check validity of traces of empirical executions.
3. **Smart contracts** are computer code that represent the terms of a contract and automatically execute at relevant events with the purpose of control or documentation [16, 27].
4. **Solid** (derived from "Social Linked Data") is a collection of technological specifications for personal online datastores (Pods) supported by authentication and authorization using Semantic Web technologies [25, 23].

Our main contributions are:

- Provision of a system architecture to support self sovereignty of personal data by combining i) decentralised technologies (Blockchain, Smart Contracts), semantic web technologies (Solid, ontologies) and a formal language (eFlint) to support data verification, proof of compliance and a method for certification and enforcement of personal consent
- Presentation of main building blocks to enable these functionalities by providing an eFlint ontology representation to create loggable traces of empirical execution of processing actions on personal data
- Showing the feasibility of our approach by using our building blocks and existing applications (eFlint Ontology, Solid, MetaMask,...) in an applied use case.

4 Related Work

Decentralised technologies, such as Blockchains have been considered a viable mechanism for access control and logging of transactions thanks to its distributed architecture. Furthermore, there are various features relevant for our use case: The immutable transaction log continuously validated ensures *transaction authenticity* [21]. Smart contracts on the blockchain can intuitively monitor, *enforce complex access control policies* [11] and provide a *dynamic approach for access control* [5].

Blockchain-based access control applications developed so far predominantly use attribute-based [14], role-based [6], and fine-grained methods [7] in addition

to generic access control schema [10]. Attribute-based Encryption access control schemas [22] such as Ciphertext-Policy Attribute-Based Encryption (CP-AEB) method [1] are deemed better in performance than rule-based access control strategies such as Access Control Lists. Access control strategies and tools built on Blockchain often use eXtensible Access Control Markup Language (XACML) [20] and Solidity and Ethereum are the most preferred language and platform for smart contracts and blockchain respectively [8]. The problem of scalability on blockchains is addressed using an architecture using multi-blockchains systems on cloud [13]. There are many different ways to represent knowledge regarding personal data management: Semantic web techniques, such as ontologies and policy languages provide suitable mechanisms to encode knowledge representation of privacy-related data management, such as Consent and compliance checking. Various ontologies have been modeled based on the GDPR legislation, such as PrOnto for legal reasoning [17], DPV [19], a community-based effort to further establish an open standard, or GConsent[18] to model consent. Other approaches include policy languages, still related to semantic web technologies, such as the SPECIAL Policy Language [3] for usage policy, but also domain-specific languages that can create enforceable code blocks, such as eFLINT [26].

Solid is a project initiated by Sir Tim Berners-Lee with the goal to give people back the control over their own personal data. Every person (or agent) can have their personal data storage on the Web, over which they have full control [23]. Technically, Solid is a collection of technological specifications for read/write Linked Data supported by authentication and authorization [25]. It builds upon the RESTful HTTP service specification of the Linked Data Platform[7] which is a W3C Recommendation. It doesn't use any centralized element and all components of the Solid architecture (Pods, identity providers, apps) can be freely chosen or self-hosted. For access control, Solid has a Web Access Control (WAC) system using the Access Control List (ACL) model [24].

5 Resources

We use already existing ontologies that define common concepts from legal basis such as GDPR or other legislation: Among them, the Data Privacy Vocabulary *DPV* is a central one, which we considered in our design. Furthermore we also derived our main concepts and applied design patterns from *DUL*[8]. *eFlint* citevan2020eflint is the domain-specific language we use for formalizing executable norm specifications to create Smart Contracts on the Etherum Blockchain. Additionally, we use Linkchain for blockchain-based verification of RDF quads, Metamask Wallet [9] as acrypto wallet to interact with the Ethereum blockchain ecosystem and Rinkby[10] testnet for testing purposes. Inrupt Solid Pod [11] is used

[7] cf., `https://www.w3.org/TR/ldp/`
[8] cf., `http://www.loa.istc.cnr.it/ontologies/DUL.owl`
[9] `https://metamask.io`
[10] `https://rinkeby.etherscan.io`
[11] `https://inrupt.com/solid`

for storing personal data. For future work, we plan to take advantage of IPFS [12] for storing and NFT Storage [13], for non-fungible tokens.

6 Proposed approach

Our architecture (see Fig. 1) consists of different components to enable data verification, compliance checking, immutability and logging of records to achieve self sovereign data management.

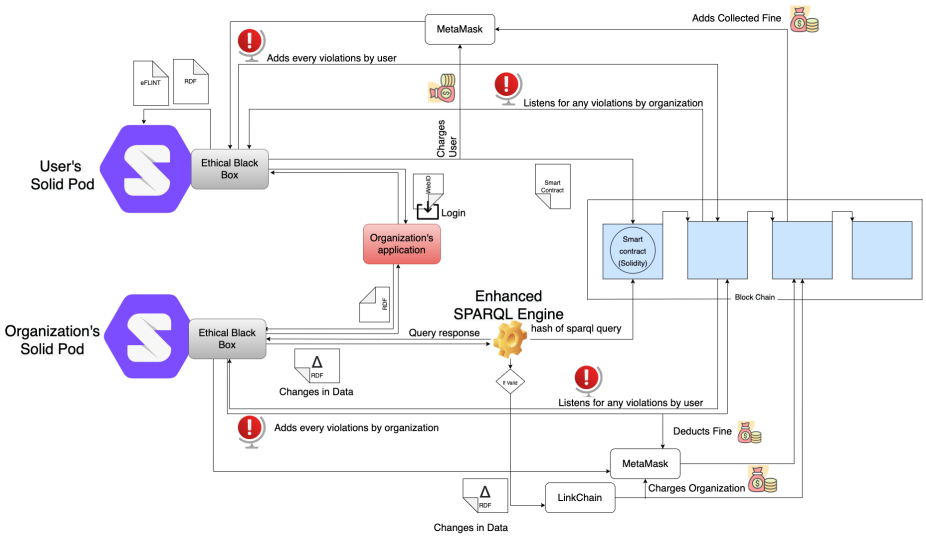

Fig. 1. Overall architecture of our approach.

EBB (EBB): The EBB is the main component of the architecture and responsible for the main tasks to achieve self soverignty in data management. The EBB is available as an extension to the Solid Pod to handle the requests for user's data on the pod. This is acheived through the application of different technologies, such as OWL representation for providing general user specifications, eFlint, a domain-specific language for executable norm specifications and Solidity for translating it to smart contracts. It also comes with an enhanced, embedded SPARQL query engine.

The main functions of the EBB include: **On the User's end**:

1. It creates the smart contract based on which data can be accessed and how, as specified by the user using eFLINT API which is then translated into

[12] `https://ipfs.io`
[13] `https://nft.storage/`

RDF representation and is then stored in the pod. Data sharing and processing specifications are described in eFLINT and has static definitions of permissions/violations, powers which are then translated into a smart contract. One would not want to replace the smart contract frequently because it would cost them for every transaction to write on the Blockchain.
2. It handles requests from the data collector and checks with the user allocated permissions on the data being requested, via the RDF file, generated previously, to verify that the data collector has the required access rights.
3. It subscribes itself to listen for violations on the blockchain and when the smart contract reports a violation, it then terminates the contract with the data collector.

On the Data Collector's end:

1. It watches for data coming from a WebID not matching the WebID of the pod that the Black Box is working on and invokes the LinkChain API to anchor the modified data to the blockchain if the enhanced SPARQL query engine verifies first that the data collector is allowed to do so.
The main function of the EBB is to enforce data recording on blockchain to monitor what is being done to the user's data on the pod.

The overall flow in a commercial setting works as follows:

1. The user logs in with their WebId into the Data Collector's portal.
2. The Data Collector then requests for the data it needs from the user.
3. The EBB creates an instance of eFLINT and creates an equivalent copy of this file in RDF and stores them on the user's pod. It will be used by the EBB to validate at regular intervals whether the data collector is within the bounds of the agreed terms later.
4. EBB checks if the user has authorized this sharing of the data and if they have approved it initiates a smart contract and the user is charged a small fee through their MetaMask account. The smart contract is anchored to the blockchain on the network.
5. If the Data Collector requests for data that the user has not given access to then, EBB asks the user if they are willing to and if agreed adds those additional terms to eFLINT and makes a new smart contract and the user is charged through MetaMask.
6. The Data Collector has access to the user's data now in their pod based on the mutually agreed terms.
7. The EBB on the Data Collector's end is now monitoring the Data Collector's pod and notices that there is new data belonging to another WebId that is not their own.
8. The EBB then invokes the Enhanced SPARQL query engine which then now checks with the smart contract on the blockchain network if the Data Collector is allowed to perform this action on the user's data and is within the permitted time-bounds and returns the result of this query to the EBB.
9. The hash of this SPARQL query and the response is anchored to the blockchain using the LinkChain API.

10. If the Data Collector is found not permitted, then the smart contract marks this as a violation and a record is created on the blockchain.
11. The EBB who is subscribed to listen on the blockchain for violations, caught by the smart contract, now removes access for that Data Collector from the access control list corresponding to the data on the user's pod and the contract is terminated.
12. If the SPARQL query result suggests that the Data Collector was indeed permitted to do what they did, then the EBB validates through the LinkChain API that the returned SPARQL response was not tampered with by the Data Collector.
13. If the result of this validation comes out clean, then the EBB via the LinkChain API deploys whatever modifications or actions were performed on the data to the blockchain at which point the transaction goes via MetaMask and the Data Collector is charged a small transaction fee which is then transferred to the user.

eFLINT to RDF Ontology: Another component of our proposed architecture is the eFLINT ontology description (cf. Fig 2). The engineering of this ontology is closely based on DUL [14], a commonly used foundational ontology. The main concepts are:

- eFlintSituation (`:eFlintSituation`): is a subclass of `DUL:Situation`, describing a general event being modeled in eFlint, this can be for example a request of the data subject to update the data (`:RectificationDemandSituation`), a controller requesting data (`:PersonalDataCollectionSituation`), the act of providing consent (`:ConsentGivingSituation`). The `eFlintSituation` can have a description (`:eFlintDescription`), an associated location (`:Place`) and validity (`:TimeInterval`).
- Consent (`:Consent`): is associated with the applicable (`:eFlintSituation`) and the Data (`:Data`) given by the data subject (`:Subject`) and given to the controller (`:Controller`). The data-related terms (`:Data`, `:Subject` and `:Controller`) can be reused from the already existing *Data Privacy Vocabulary* (DPV)[15].
- Data (`:Data`) can be specified more detailed into different types (`:PersonalDataType`) and has is accurate for certain purposes (`:Purpose`). Also in this case, applicable terms from DPV can be reused.

7 Evaluation and Results: Use case/Proof of concept - Experiments

In this section we discuss our evaluation use case: Alice, the user wants to manage her health data and consent cf. Fig. 3. Alice wants to enforce the following

[14] http://www.loa.istc.cnr.it/ontologies/DUL.owl
[15] cf., `https://w3c.github.io/dpv/dpv/`

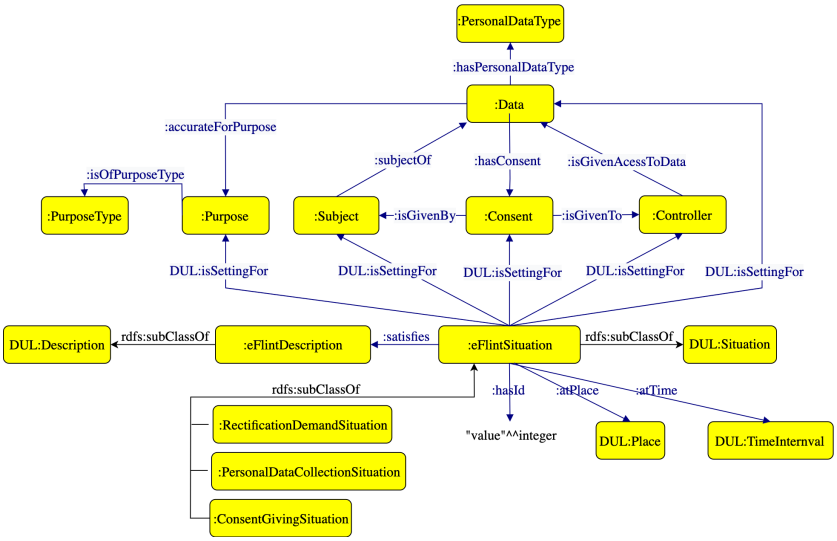

Fig. 2. The general pattern of the eFlint situation, following the DnS pattern from DUL.

natural language consent: *"I give consent for "Health App" to collect my (reproductive system-related) personal medical data, but only for non-commercial and non-governmental purposes."* Her choice of consent needs to be formalised for which we have chosen eFLint, as a turing-complete formal language. To further trace data processing and use, the eFlint ontology with the traced hashes of transactions can help to achieve empirical semantics in a privacy-preserving way. To this end, from the resulting eFLINT-ontology-complaint RDF file based on her consent choices, an analogous eFlint file can be produced. This eFlint file then can be translated into a Smart Contract in Solidity to be deployed on the Etherum Blockchain to have an enforceable, automatic compliance checking mechanism. For now, we have manually checked the feasibility of our approach by creating the necessary files and translation mappings but for future work, we want to create the applicable system endpoints to automatically create the necessary files.

8 Discussion and Conclusions

Current approaches of data management focus on centralised solutions leaving the user unaware of data exploitation and violations t. Furthermore, there is little evidence on how and what data is used for different use cases and contexts. Decentralised technologies can offer an alternative to enable users to enforce increased governance on their personal data and also introduce empirical evidence by tracking data usage. With our work we want to make first steps towards self

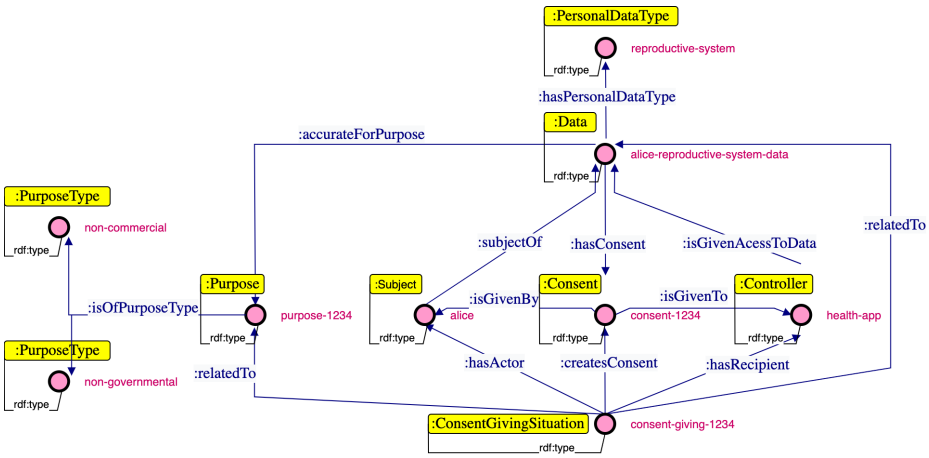

Fig. 3. Instantiating the use case for the eFlintSituation

sovereignty in terms of personal data management by combining decentralised technologies, ontologies and formal languages.

RDF and ontologies can be used to give the explicit semantics of the duties, desires, consent, etc from the user to eFlint. Then, compared to other access control strategies (Access Control Lists), smart contracts on blockchain can intuitively handle time-bound access for Data. It can ensure complex access control policies with the help of tools like eFLINT.

References

[1] John Bethencourt, Amit Sahai, and Brent Waters. "Ciphertext-Policy Attribute-Based Encryption". In: *2007 IEEE Symposium on Security and Privacy (SP '07)*. 2007, pages 321–334. DOI: 10.1109/SP.2007.11.

[2] L. Thomas van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. "EFLINT: A Domain-Specific Language for Executable Norm Specifications". In: GPCE 2020. Virtual, USA: Association for Computing Machinery, 2020, 124â136. ISBN: 9781450381741. DOI: 10.1145/3425898.3426958.

[3] Piero A Bonatti, Sabrina Kirrane, I Petrova, L Sauro, and E Schlehahn. *SPECIAL Policy Language V2*. Technical report. 2018. URL: https://www.specialprivacy.eu/images/documents/SPECIAL%7B%5C_%7DD25%7B%5C_%7DM21%7B%5C_%7DV10.pdf.

[4] Katrina Brooker. *"I was devastated": The man who created the World Wide Web has some regrets*. July 2018. URL: https://www.vanityfair.com/news/2018/07/the-man-who-created-the-world-wide-web-has-some-regrets.

[5] Seraphin Calo, Dinesh Verma, Supriyo Chakraborty, Elisa Bertino, Emil Lupu, and Gregory Cirincione. "Self-Generation of Access Control Policies". In: *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*. SACMAT '18. New York, NY, USA: Association for Computing Machinery, June 2018, pages 39–47. ISBN: 978-1-4503-5666-4. DOI: 10.1145/3205977.3205995. URL: https://doi.org/10.1145/3205977.3205995 (visited on 07/06/2022).

[6] Gaby G Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology". In: *Sustainable cities and society* 39 (2018), pages 283–297.

[7] Gaby G. Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology". en. In: *Sustainable Cities and Society* 39 (May 2018), pages 283–297. ISSN: 2210-6707. DOI: 10.1016/j.scs.2018.02.014. URL: https://www.sciencedirect.com/science/article/pii/S2210670717310685 (visited on 07/06/2022).

[8] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. "Blockchain Based Access Control". In: *17th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS)*. Edited by Lydia Y. Chen and Hans Reiser. Volume LNCS-10320. Distributed Applications and Interoperable Systems. Neuchatel, Switzerland: Springer International Publishing, June 2017, pages 206–220. DOI: 10.1007/978-3-319-59665-5_15. URL: https://hal.inria.fr/hal-01800124 (visited on 07/06/2022).

[9] Beth Duff-Brown. *Protecting reproductive health information after fall of Roe v. Wade*. June 2022. URL: https://law.stanford.edu/press/protecting-reproductive-health-information-after-fall-of-roe-v-wade/.

[10] Shuang Hu, Lin Hou, Gongliang Chen, Jian Weng, and Jianhua Li. "Reputation-Based Distributed Knowledge Sharing System in Blockchain". In: *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. MobiQuitous '18. New York, NY, USA: Association for Computing Machinery, 2018, 476â481. ISBN: 9781450360937. DOI: 10.1145/3286978.3286981. URL: https://doi.org/10.1145/3286978.3286981.

[11] *Introduction to smart contracts*. URL: https://ethereum.org/en/smart-contracts/ (visited on 07/08/2022).

[12] Issie Lapowsky. *How Cambridge Analytica sparked the Great Privacy Awakening*. Mar. 2019. URL: https://www.wired.com/story/cambridge-analytica-facebook-privacy-awakening/.

[13] Mingxin Ma, Guozhen Shi, and Fenghua Li. "Privacy-Oriented Blockchain-Based Distributed Key Management Architecture for Hierarchical Access Control in the IoT Scenario". In: *IEEE Access* 7 (2019). Conference Name:

IEEE Access, pages 34045–34059. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2904042.

[14] Damiano Di Francesco Maesa, Paolo Mori, and Laura Ricci. "A blockchain based approach for the definition of auditable access control systems". In: *Computers & Security* 84 (2019), pages 93–119.

[15] Ralph C. Merkle. "A Digital Signature Based on a Conventional Encryption Function". In: *Advances in Cryptology — CRYPTO '87*. Edited by Carl Pomerance. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pages 369–378. ISBN: 978-3-540-48184-3.

[16] Nick Szabo. *Smart Contracts*. 1994. URL: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (visited on 07/08/2022).

[17] Monica Palmirani, Michele Martoni, Arianna Rossi, Cesare Bartolini, and Livio Robaldo. "Pronto: Privacy Ontology for Legal Compliance". In: *Proc. 18th European Conference on Digital Government (ECDG)*. 2018, pages 142–151.

[18] Harshvardhan J Pandit, Christophe Debruyne, Declan OâSullivan, and Dave Lewis. "GConsent-a consent ontology based on the GDPR". In: *European Semantic Web Conference*. Springer. 2019, pages 270–282.

[19] Harshvardhan J Pandit, Axel Polleres, Bert Bos, Rob Brennan, Bud Bruegger, Fajar J Ekaputra, Javier D Fernández, Roghaiyeh Gachpaz Hamed, Elmar Kiesling, Mark Lizar, et al. "Creating a vocabulary for data privacy". In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer. 2019, pages 714–730.

[20] Erik Rissanen. *eXtensible Access Control Markup Language (XACML) Version 3.0*. OASIS Standard. Jan. 2013. URL: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html (visited on 07/06/2022).

[21] Sara Rouhani and Ralph Deters. "Blockchain based access control systems: State of the art and challenges". In: *IEEE/WIC/ACM International Conference on Web Intelligence*. Oct. 2019, pages 423–428. URL: http://arxiv.org/abs/1908.08503 (visited on 07/06/2022).

[22] Amit Sahai and Brent Waters. "Fuzzy identity-based encryption". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2005, pages 457–473.

[23] Andrei Vlad Sambra, Essam Mansour, Sandro Hawke, Maged Zereba, Nicola Greco, Abdurrahman Ghanem, Dmitri Zagidulin, Ashraf Aboulnaga, and Tim Berners-Lee. *Solid: A Platform for Decentralized Social Applications Based on Linked Data*. en. Technical Report. MIT CSAIL & Qatar Computing Research Institute, 2016. (Visited on 11/10/2020).

[24] Sarven Capadisli. *Web Access Control*. en. July 2021. URL: https://solidproject.org/TR/wac (visited on 07/08/2022).

[25] Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, and Kjetil Kjernsmo. *Solid Protocol*. Dec. 2021. URL: https://solidproject.org/TR/protocol (visited on 06/03/2022).

[26] L Thomas Van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. "eFLINT: a domain-specific language for executable norm specifications". In: *Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*. 2020, pages 124–136.

[27] Vitalik Buterin. "A Next-Generation Smart Contract and Decentralized Application Platform". en. Whitepaper. 2014. URL: `https://ethereum.org/en/whitepaper/` (visited on 07/08/2022).

9 Appendix

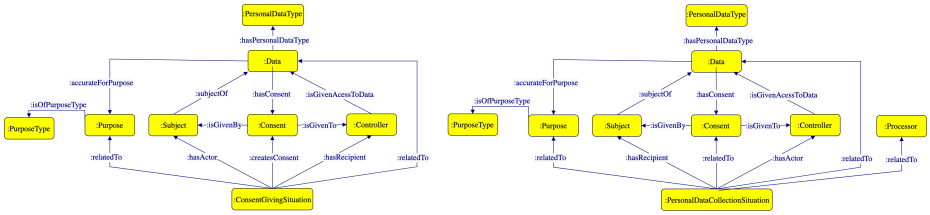

Fig. 4. Two specializations of the general eFLINT situation. Left: the consent giving situation. Right: the data processing situation.

```
Fact subject
Fact controller
Fact personal-data-type
Fact purpose
Fact data Identified by subject * personal-data-type
Fact consent Identified by subject * controller * data * purpose
Fact accurate-for-purpose Identified by data * purpose

Act gives-consent
    Actor subject
    Recipient controller
    Related to data, purpose
    Creates consent()
    Holds when !consent()

+subject(alice).
+controller(health-app).
+personal-data-type(reproductive-system).
+purpose(non-commercial-non-governmental).
+data(alice, reproductive-system).

!gives-consent(alice, health-app, data(alice, reproductive-system), non-commercial-non-
governmental).
```

Fig. 5. The eFLINT specification of Fig.3 and Fig.4.