decode

Privacy Design Strategies for the DECODE Architecture

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 732546

Project no. 732546

DECODE

DEcentralised Citizens Owned Data Ecosystem

D1.2 Privacy Design Strategies for the DECODE Architecture

Version Number: V1.0

Lead beneficiary: RU

Due Date: June 2017

Author(s): Shehar Bano (UCL), Eleonora Bassi (POLITO), Marco Ciurcina (POLITO), Ana Freire (EURECAT), Sara Hajian (EURECAT), Jaap-Henk Hoepman (RU)

Editors and reviewers: Denis Roio (Dyne), Priya Samuel (TW), Francesca Bria (IMI)

Dissemination level:		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Approved by: Francesca Bria, IMI

Date: 30/06/2017

This report is currently awaiting approval from the EC and cannot be not considered to be a final version.

Table of Contents

Introduction

DECODE aims to develop a privacy preserving data distribution platform to foster commons-based sharing economy models, where citizens own and control their data. This asks for a privacy by design-based approach, for which the concept of privacy design strategies have recently been developed.

The General Data Protection Regulation (GDPR), as well as other data protection or privacy protection laws and regulations, define data protection in legal terms. These terms are soft, open to interpretation, and highly dependent on context. Because of this inherent vagueness, engineers find such legal requirements hard to understand and interpret.

The GDPR also mandates privacy by design, without describing clearly what this means exactly, let alone giving concrete guidelines on how to go about implementing privacy by design when actually designing a system. Intuitively, privacy design means addressing privacy concerns throughout the system development lifecycle, from the conception of a system, through its design and implementation, proceeding through its deployment all the way to the decommissioning of the system many years later. In terms of software engineering, privacy is a quality attribute, like security, or performance. To make privacy by design concrete, the soft legal norms need to be translated into more concrete design requirements that engineers understand. This is achieved using privacy design strategies.

In this deliverable, we describe the legal constraints (2), describe the initial DECODE architecture (3), describe and apply the privacy design strategies approach to it (4) and discuss privacy in relation to non-discriminatory data mining in the context of DECODE (5). The result is a list of concrete recommendations to guide the design and implementation of the DECODE architecture. This deliverable is structured accordingly.

Legal constraints and prescribed methods

According to Privacy and Data Protection European legislation, privacy by design measures should assist all the data processing phases (i.e. architecture and data processing) in order to protect and enhance individual rights and the ethical coherence of the entire project (DECODE).

General Legal Framework on Privacy by Design:

Since the late '90 the principle of privacy by design was introduced by Ann Cavoukian (Cavoukian, 2010) and its fortune in European legal framework is stated from the document "The Future of Privacy" (02356/09/EN – WP168) adopted on December 1st, 2009 by EU Article 29 Data Protection Working Party (WP29) and the Working Party on Police and Justice (WPPJ).

In 2012 it was included in the Proposal of revision of the Directive 95/46/EC, and, finally, it was fixed in Article 25 of the General Data Protection Regulation (GDPR), Regulation (EU) 2016/679, that introduces legal obligation to design strategies.

For a full comprehension of the implications of the PbD principle, it should be interpreted in accordance with the recommendations by Working Party Art. 29 and by the European Data Protection Supervisor (see EDPS opinion on privacy in the digital age: "Privacy by Design" as a key tool to ensure citizens' trust in ICTs), and taking advantages from the standards and principles stressed by the International Standard Organization (ISO 29100).

The GDPR applies from 25 May 2018 and replaces Directive 95/46/EC.

According to Article 25 of GDPR "Data protection by design and by default":

> 1. Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, such as pseudonymisation, which are designed to implement data-protection principles, such as data minimisation, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects.
>
> 2. The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. That obligation applies to the amount of personal data collected, the extent of their processing, the period of their storage and their accessibility. In particular, such measures shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons.
>
> 3. An approved certification mechanism pursuant to Article 42 may be used as an element to demonstrate compliance with the requirements set out in paragraphs 1 and 2 of this Article.

Who is responsible to adopt privacy by design measures?

According to Article 25, the adoption of privacy by design measures is clearly an obligation of the data controller (see Articles 4, 1 (7) and 24, GDPR, and also Opinion 1/2010 on the concepts of "controller"

and “processor”, adopted by WP29 (wp169)). Moreover, the distributed architecture of Decode requires to stress carefully how to individuate the data controller, or the joint data controllers (according to Article 26, GDPR “Joint controllers” with regarding to the determination of the means of the processing). This norm has a strong impact on the transparency of the processing (on transparency see also De Filippi, 2015?):

> 1.Where two or more controllers jointly determine the purposes and means of processing, they shall be joint controllers. They shall in a transparent manner determine their respective responsibilities for compliance with the obligations under this Regulation, in particular as regards the exercising of the rights of the data subject and their respective duties to provide the information referred to in Articles 13 and 14, by means of an arrangement between them unless, and in so far as, the respective responsibilities of the controllers are determined by Union or Member State law to which the controllers are subject. The arrangement may designate a contact point for data subjects.
>
> 2. The arrangement referred to in paragraph 1 shall duly reflect the respective roles and relationships of the joint controllers vis-à-vis the data subjects. The essence of the arrangement shall be made available to the data subject.
>
> 3. Irrespective of the terms of the arrangement referred to in paragraph 1, the data subject may exercise his or her rights under this Regulation in respect of and against each of the controllers.

Recital 78 of the GDPR and scholars (for instance Koops & Leenes, 2014) outline that the adoption of privacy by design measures is not only an obligation for data controllers, but also a recommendation for IT systems producers.

> (...) When developing, designing, selecting and using applications, services and products that are based on the processing of personal data or process personal data to fulfill their task, producers of the products, services and applications should be encouraged to take into account the right to data protection when developing and designing such products, services and applications and, with due regard to the state of the art, to make sure that controllers and processors are able to fulfill their data protection obligations. The principles of data protection by design and by default should also be taken into consideration in the context of public tenders. (GDPR, Rec. (78))

These considerations require to stress roles and responsibilities of different nodes of Decode’s architecture (see Berberich & Steiner, 2016).

When should privacy by design measures be adopted?

Privacy by design technical and organizational measures shall be adopted “both at the time of the determination of the means for processing and at the time of the processing itself” (Article 25, GDPR). This distinction aims to underline the necessity of pursuing privacy preserving goals in the design phase and in all the phases of the processing, introducing a sort of double responsibility both in design and implementation. So that, to be compliant to this provision, it seems necessary to define and design the architecture both of the entire system and the data processing, in order to be able to individuate critical phases and requirements, and then to choose which are the measures to implement.

What kind of measures do we decide to adopt?

Article 25 mentions pseudonymisation and data minimisation as examples of privacy by design measures, but the list is longer and includes security measures, encryption, anonymisation, aggregation, third parties limitation access, tools for ensuring data subject's informed consent and data subject's right (among others: the right to erasure, the right to be forgotten and data portability).

Some measures could be described as self-enforceable, while others are directed to conduct or to change user’s behaviour (Pagallo, 2012). Moreover, some scholars distinguish between measures adopted by code and measures adopted by policy (or by communication (see for instance Koops &

Leenes, 2014)). Similarly, in DECODE it is also used the distinction between measures adopted by cryptography and measures adopted by convention.

Nonetheless it seems that all these measures are not alternative but compatible. These are strategies, patterns and tools that can work simultaneously to gain specific goals in preserving and in enforcing privacy rights (see Hoepman, 2014; Colesky et al., 2016), following the idea of modularity of design.

High level architecture description

DECODE provides a distributed and privacy-aware architecture for decentralised data governance and federated identities. Key components of this architecture are as follows:

DECODE Hubs and Nodes

A device that runs the DECODE OS is called the DECODE HUB. The DECODE NODE is an interface that can be used to configure smart rules. It also abstracts common choices regarding granting access to data and mapping its use. The DECODE OS provides a cross-platform, securely connected base operating system that grants the integrity of execution of the NODE rules and applications. It can run on different hardware (e.g., a GNU/Linux/BSDbased computer, a mobile device, or a file server) or can be virtualized inside other host operating systems.

Smart Rules

DECODE's fully decentralised architecture offers a flexible and extensible data governance, which enables fine-grained control of different regimes of data ownership and privacy. Smart rules are a set of algorithmic protocols expressed in a formal language that implement this flexibility. Data owners can use smart rules to define how data should be managed in terms of access, value attribution and other parameters, and legal/contractual obligations and other constraints. Smart rules follow a defined ontology to define access to subsets of data (e.g., personal data or for specific uses granted to specific subjects). Entitlements have a lifetime and are only valid for a certain period. Smart rules can also be used to revoke authorisation for access or change the legal status and the conditions of use and exploitation of the data.

Smart rules can be expressed in a declarative language (which may be visually represented), which is then compiled in a functional language and executed. Executed smart rules can provide basic functions of governance and identity management: publish/subscribe access to events and functions to interface with external APIs; core functions to store and access the blockchain according to Attribute Based Cryptography entitlements; library functions to interfaces with external applications.

Smart rules enable providers and app developers to define rules about operation of the system or the regulatory environment. Such abstraction between people's choices and its enforcement creates a rich landscape for flexible and decentralised creation of new applications and services.

The smart rules, as well as all the platform specifications, protocols, ontology, semantic specifications, will be released under a Free and Open Source Software (FOSS) license. The initial set of rules will be gradually extended with community participation as requirements evolve, with the goal to eventually emerge as the standard language for managing data access and valorization in distributed and decentralised architectures.

Distributed Ledger

A distributed ledger is a decentralized data repository that is resistant to malware and hacking, and provides privacy and transparency through ABC (Attribute Based Cryptography) and other privacy-enhancing technologies. Cryptographic primitives enforce strict access control that maintain the ledger's security and accuracy. A fully decentralized platform is realized by combining smart rules with distributed ledger technologies, which enforces by design flexible and extensible data governance.

The platform supports multiple, diverse contexts of data ownership and privacy. A heterogeneous set of data streams can be collected and fed to the platform: civic datasets and open linked data, private data, personal data with undisclosed identity and personal data associated with an electronic identity.

Data confidentiality is enforced by encryption, which also allows anonymization since association with an e-identity depends on permissions specified by the data owner. Thus personal data can be consensually and anonymously used for collective intelligence, or for personalised services and applications (if authorised by data subjects). User authorizations are managed by defining ontologies and indexes over data streams collected by sensors, IoT objects or personal devices.

We implement the distributed ledger as Chainspace[1]—a distributed ledger platform for high-integrity and transparent processing of transactions within a distributed or decentralized system. Unlike application specific distributed ledgers, such as Bitcoin (Nakamoto, 200*) supporting a currency, or certificate transparency (Laurie et al, 2013) supporting certificate verification, Chainspace offers extensibility though supporting smart contracts, like the Ethereum platform (Wood, 2014). However, the Chainspace system is exposed to enough information about contract and transactions, in order to support and provide higher scalability through automatic sharding. The platform is agnostic as to the smart contract language, or identity infrastructure. Privacy features can be integrated in the system through modern zero-knowledge proofs or SNARKs.

[1] See https://gogs.dyne.org/DECODE/wip/src/5477da0ac089795ecc13c11b64a3591e9cfae121/distributed-ledger/ChainspaceDL_april17.pdf

Privacy design strategies (RU)

As explained in the introduction, the GDPR defines data protection in more vague legal terms. Engineers find such legal requirements hard to understand and interpret. In particular, the GDPR also mandates privacy by design, without describing clearly what this means exactly, let alone giving concrete guidelines on how to go about implementing privacy by design when actually designing a system.

To make privacy by design concrete, the soft legal norms need to be translated into more concrete design requirements that engineers understand. And tools to elicit and implement these requirements need to be available. For the design and implementation phase, such tools are available. In particular, so called *privacy enhancing technologies* (PETS) have been developed in the last thirty years ago (starting with the seminal work of David Chaum in the eighties). And also for the design phase, privacy design patterns have started to emerge. (We will discuss them briefly further on in this report). Unfortunately, until recently concrete tools to address privacy during the early design phases of a system, i.e. during the concept formulation and analysis phase, were missing. This is why *privacy design strategies* have been developed.

As described in (Colesky et. al. 2016) a privacy design strategy specifies a distinct architectural goal in privacy by design to achieve a certain level of privacy protection. It is noted that this is different from what is understood to be an architectural strategy within the software engineering domain. Instead our strategies can be seen as goals of the privacy protection quality attribute (where a quality attribute is a term from software engineering describing non-functional requirements like performance, security, and also privacy).

In the description of the privacy design strategies we frequently refer to *processing* of personal data. Engineers[2] should be aware that the legal concept of processing is broader than what a typical engineer understands processing to mean. In what follows we use the legal interpretation of processing, which includes creating, collecting, storing, sharing and deleting personal data.

[2] When writing “engineer” we mean the large class of professionals that are tasked to engineer something. In particular this includes “developers”, “systems developers”, “software developers” etc.

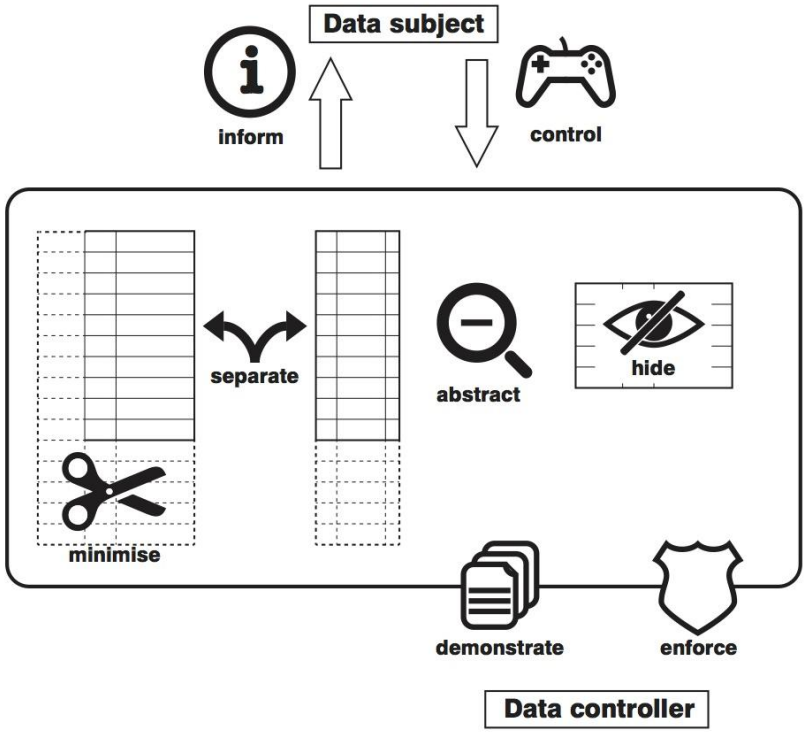

Figure 1 Privacy design strategies

We now proceed to briefly describe the eight privacy design strategies (see figure 1). More information can be found in (Hoepman, 2014 and Colesky et al, 2016). Each strategy is first described using a brief definition. Subsequently, the strategy is refined by one or more *tactics* that each describe a different way in which the overarching strategy can be achieved. We then present some examples by showing how the strategy impacts the DECODE architecture.

Minimise

Definition: Limit the processing of personal data as much as possible.

Associated tactics:

Exclude: refraining from processing a data subject's personal data, partly or entirely, akin to blacklisting or opt out.

Select: decide on a case by case basis on the full or partial usage of personal data, akin to whitelisting or opt-in.

Strip: removing unnecessary personal data fields from the system's representation of each user.

Destroy: completely removing a data subject's personal data.

Example / impact on DECODE:

Explicit selection or exclusion of data items should be done by de DECODE nodes according to strict rules, that depend on the specific application at hand.

As an example to satisfy the "*destroy*" tactic, data items processed by DECODE should always be tagged with an expiry date. This should happen at the smallest granularity of data items where a distinction between expiry dates is relevant. Any node that encounters a data item whose expiry date lies in the past should discard that data item (and any copies thereof under that node'' control). This can also be enforced by the smart rules supported by the DECODE architecture.

The minimise strategy is also captured by 'datensparsamkeit'[3], a German word that is hard to translate into English. It is a concept from privacy laws that is an opposite philosophy to "capture-all-the-things".

Separate

Definition: Prevent correlation of personal data by separating the processing logically or physically

Associated tactics:

Distribute: partitioning personal data so that more access is required to process it.

Isolate: processing parts of personal data independently, without access or correlation to related parts.

Example / impact on DECODE:

DECODE's use of a federated distributed storage based on blockchain technology, where the DECODE nodes each independently contribute their resources (storage and compute cycles) in a controlled manner to the overall system implements the "*distribute*" tactic.

DECODE in the end provides a platform on top of which many different applications may run simultaneously. Isolation can be achieved by tagging data items with the application they were collected or created for, to avoid reuse of data items for other applications. The smart rules (that typically are at the core of such applications) can be used to force this isolation (by checking the tags of all data items they process), but we note this is a type of isolation 'by convention' only that in theory can be bypassed,

Abstract

Definition: Limit as much as possible the amount of detail of personal data being processe

Associated tactics:

Summarise: extracting commonalities in personal data by finding and processing correlations instead of the data itself.

Group: inducing less detail from personal data prior to processing, by allocating into common categories.

Perturb: add noise or approximate the real value of a data item.

Example / impact on DECODE:

There may be a way for DECODE to apply the Summarise tactic if it provides a query engine for aggregated data that removes all details and only provides aggregated information. Perturbation can perhaps be used in an ad-hoc fashion to the storage of raw data in some cases.

The use of this strategy can also be explored within the Decidim Petitions. When users are signing a petition, any additional data such as their date of birth, postal code, gender can be abstracted as groupings or aggregations of data instead of gathering raw data.

General examples of this strategy are recording the age instead of date of birth, postal code instead of specific address, etc.

[3] https://martinfowler.com/bliki/Datensparsamkeit.html

Hide

Definition: protect personal data, or make them unlinkable or unobservable. Prevent personal data becoming public. Prevent exposure of personal data by restricting access, or hiding its very existence.

Associated tactics:

Restrict: preventing unauthorized access to personal data.

Mix: processing personal data randomly within a large enough group to reduce correlation.

Encrypt: encrypt data (in transit or at rest)

Obfuscate: preventing understandability of personal data to those without the ability to decipher it.

Dissociate: removing the correlation between different pieces of personal data.

Example / impact on DECODE:

Within DECODE access to data, including to personal data, is restricted based on the concept of entitlements. Only an actor possessing the necessary entitlements can access a resource. Entitlements are implemented based on the concept of attribute based credentials, that in fact provide another layer of privacy protection due to the fact that they are unlinkable to individuals (unless, of course, the attributes used identify a person by name or number). Finally, all personal data in DECODE is encrypted.

Inform

Definition: provide data subjects with adequate information about which personal data is processed, how it is processed, and for what purpose.

Associated tactics:

Supply: making available extensive resources on the processing of personal data, including policies, processes, and potential risks.

Notify: alerting data subjects to any new information about processing of their personal data in a timely manner.

Explain: detailing information on personal data processing in a concise and understandable form.

Example / impact on DECODE:

DECODE aims to implement this strategy in several ways. First of all, there is a strong focus on open source hard and software, both for the DECODE nodes as the interconnecting infrastructure. Second, transparency and auditability are core technical values. This is exemplified by the use of open blockchain technology for the mediation of all transactions involving personal data. Finally, DECODE aims to deploy declarative and intelligible smart-rules that can be well related to legal taxonomy.

Control

Definition: provide data subjects mechanisms to control the processing of their personal data.

Associated tactics:

Consent: only processing the personal data for which explicit, freely-given, and informed consent is received.

Choose: allowing for the selection or exclusion of personal data, partly or wholly, from any processing.

Update: providing data subjects with the means to keep their personal data accurate and up to date.

Retract: honouring the data subject's right to the complete removal of any personal data in a timely fashion.

Example / impact on DECODE:

The main element of control provided to users of the DECODE platform is by expressing entitlement conditions for the access to individual pieces of personal data. A graphical user interface and an intuitive language to express these conditions and the associated entitlements and smart rules will be developed to support the user to exert his or her control.

Enforce

Definition: commit to a privacy friendly way of processing personal data, and enforce this.

Associated tactics:

Create: acknowledging the value of privacy and deciding upon policies which enable it, and processes which respect personal data.

Maintain: considering privacy when designing or modifying features, and updating policies and processes to better protect personal data.

Uphold: ensuring that policies are adhered to by treating personal data as an asset, and privacy as a goal to incentivize as a critical feature.

Example / impact on DECODE:

Services provided over the DECODE infrastructure should specify a clear privacy policy. The DECODE infrastructure itself should provide tools for easy enforcement of such privacy policies. This is partially achieved through the concept of entitlements, matching attributes, and the use of smart rules that govern the access to personal data.

Demonstrate

Definition: provide evidence that you process personal data in a privacy friendly way.

Associated tactics:

Log: tracking all processing of data, without revealing personal data, securing and reviewing the information gathered for any risks.

Audit: examining all day to day activities for any risks to personal data, and responding to any discrepancies seriously.

Report: analysing collected information on tests, audits, and logs periodically to review improvements to the protection of personal data.

Example / impact on DECODE:

So-called Privacy Impact Assessments (PIA), also for the cases in which is not mandatory, can be a good tool for assuring certainty, transparency of the processing and fostering trust in the architecture (mainly for public sector bodies and companies who are committed to adopt a PIA). It is recommended to each service provider offering services over the DECODE architecture to perform such a PIA.

The DECODE architecture itself facilitates this through the use of open blockchain technology for the mediation of all transactions involving personal data, thus providing accountability as a built-in feature.

Privacy and non-discriminatory data mining

Living in the information society facilitates the automatic collection of huge amounts of data on individuals, organizations, etc. Publishing, mining and personalising of high data quality data for secondary analysis (e.g. learning models, finding patterns, personalise services) may be extremely useful to policy makers, planners, marketing analysts, researchers and others. Yet, data publishing and mining do not come without dangers, namely privacy invasion and also potential discrimination of the individuals whose data are published and analysed.

Note that, some of the data mining and recommender systems may run on end-user's personal device and sharing personal data with a company/third party is not required and thus, a secure analysis is achievable. However, in the most of the cases the data mining and recommender system approaches are different because a big collection of end-user's data has to be considered to create one single recommendation or decision. As a consequence, companies maintain huge databases for end-user data. The problem is how to design privacy-preserving solutions for data mining and recommender system to ensure the privacy issues:

1. Data ownership. Ensuring that data subjects/users own and control their personal data. As such, the system is designed by default to recognize end-users as the owners of data and the mining and recommendation services as guests with delegated permissions.
2. Data transparency and auditability. Each end-user has complete transparency over what data is being collected about her and how they are accessed.
3. Fine-grained access control. At any given time, the end-user may alter the set of permissions and revoke access to previously collected data.

Current solutions for achieving privacy preserving data mining and recommendation systems

There have been various attempts to address the above privacy risks, not only from a legislative perspective, but also from a technological perspective:

1. Data anonymization methods attempt to protect personally identifiable information. k-anonymity, a common property of anonymized datasets requires that sensitive information of each record is indistinguishable from at least k−1 other records [Sweeney, 2002]. Related extensions to k-anonymity include l-diversity, which ensures the sensitive data is represented by a diverse enough set of possible values [Machanavajjhala et al., 2007]; and t-closeness, which looks at the distribution of sensitive data [Li et al., 2007]. Recent research has demonstrated how anonymized datasets employing these techniques can be de-anonymized [Narayanan et al., 2006], [Montjoye et al., 2013], given even a small amount of data points or high dimensionality data.
2. Differential privacy, a technique that perturbs data or adds noise to the computational process prior to sharing the data [Dwork, 2006]. The main drawback of these technique is that the utility of the perturbs data is low.
3. Encryption schemes that allow running computations and queries over encrypted data. Specifically, fully homomorphic encryption (FHE) [Gentry, 2009] schemes allow any computation to run over encrypted data, but are currently too inefficient to be widely used in practice.
4. Secure multiparty computation also known as distributed privacy-preserving data mining [Lindell et al., 2009]. In this scheme data are split into pieces, and shared among a distributed network of nodes or data owners. Computations are performed interactively and collaboratively on between the data owners. Thereby, individual data owner doesn't not get access to meaningful raw data,

but only on encrypted shards or the final results of the computation. The approaches also have high communication and computation complexity.

5. Block-chain supported secure multiparty computation. In recent years, a new class of accountable systems emerged. The first such system was Bitcoin, which allows users to transfer currency (bitcoins) securely without a centralized regulator, using a publicly verifiable open ledger (or blockchain). Since then, other projects (collectively referred to as Bitcoin 2.0 [Evans, 2014]) demonstrated how these blockchains can serve other functions requiring trusted computing and auditability. A novel blockchain-based approach [Zyskind et al., 2015] is able to cryptographically guarantee the proper usage of personal data. The core component is a decentralized peer-to-peer network that allows storing encrypted data in a tamper-proof way and runs secure computations while no one but the data owner has access to the data.

The main features of a blockchain, an immutable public log, and a programmable token of value, have been used to advance secure multiparty computation systems in terms of fairness and operational efficiency. Enigma [Zyskind et al., 2015] implements those advancements in order to provide an open decentralized network for encrypted data storage and secure multiparty computations. Identity, access and rules management is facilitated by the underlying protocol. Private ruls provide the programming interface to access private and public data and to specify the computations. Thereby end users can permit and audit the usage of their data in fine granularity. Moreover, they can revoke the permission at any time.

Our proposal for decentralised privacy-preserving mining and non-discriminatory recommendation[4]

Our aim is to propose a secure and private data mining and recommender system using the advantages of blockchain-supported secure multiparty computation. Using this approach end-users are able to receive personalized recommendations or the results of data mining models without disclosing their data to anyone or access to the data of others. This approach allows running computations directly on the network and obtain the final results. If data splits into shares, rather than encrypting them, we can study how to use secure Multiparty Computation (MPC) to securely evaluate any function [Lindell et al., 2009]. In this way, end-users never lose control of their data and is able to terminate the business relationship at any time. Fraud and misuse is no longer possible, because the involved company never gets the raw data.

[4] WP3 Blockchain for decentralised data and digital identity management - task 3.3

Conclusions

We have outlined the privacy issues relevant to DECODE, based on an initial sketch of the architecture. Our most important findings are the following:

1. End-users should be able to receive personalized recommendations or the results of data mining models without disclosing their data to anyone or access to the data of others.
2. Verifiers should be able to validate transactions without learning secrets and confidential data within the transaction.
3. Our initial analysis based on the privacy design strategies shows that the initial DECODE architecture is promising in the inherent privacy preserving properties it exhibits.

We recommend the following:

1. Use the advantages of blockchain-supported secure multiparty computation, in order to design a secure and private data mining and recommender system.
2. When refining the DECODE architecture in more detail, one needs to take the observations made in section 4 into account. Especially, some effort needs to be spent deciding how to address the inform, control, enforce and demonstrate strategies.
3. DECODE supports user-defined smart rules that encode the 'business' logic of specific applications. We recommend that to support privacy-friendly contracts, the design should employ mechanisms for verifiers to check validity of smart contracts without having to learn private/confidential state within the contracts.

We will update this document once the DECODE architecture has been described and decided upon in more detail, to reflect the changes in our assessment.

References

[Berberich & Steiner, 2016] M. Berberich, M. Steiner, Blockchain Technology and the GDPR – How to Reconcile Privacy and Distributed Ledgers?, EDPL (3) 2016, pp. 422-426.

[Colesky et al., 2016] M. Colesky, J-H. Hoepman, C. Hillen, A critical analysis of Privacy Design Strategies, 2016 IEEE Security and Privacy Workshops, pp. 33-40.

[De Filippi, 2016] P. De Filippi, The Interplay between Decentralization and Privacy: The Case of Blockchain Technologies. Journal of Peer Production, 2016, Issue n.7: Alternative Internets . Available at SSRN: https://ssrn.com/abstract=2852689

[Dwork, 2006] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[Evans, 2014] J. Evans. Bitcoin 2.0: Sidechains and ethereum and zerocash, oh my!, 2014.

[Gentry, 2009] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[Hoepman, 2014] J-H. Hoepman, Privacy Design Strategies, IFIP TC11 29th Int. Conf. on Information Security (IFIP SEC 2014), pp. 446-459.

[Koops & Leenes, 2014] B-J. Koops, R. Leenes, Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design' provision in data-protection law', 28 International Review of Law, Computers & Technology (2), 2014, pp. 159-171.

[Laurie et al., 2013] Ben Laurie, Adam Langley, and Emilia Kasper. Certificate transparency. Technical report, 2013.

[Li et al., 2007] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, volume 7, pages 106–115, 2007.

[Lindell et al., 2009] Y. Lindell and B. Pinkas. "Secure multiparty computation for privacy-preserving data mining." Journal of Privacy and Confidentiality 1.1 (2009): 5.

[Machanavajjhala et al., 2007] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k- anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[Montjoye et al., 2013] Y. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[Nakamoto, 2008] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[Narayanan et al., 2006] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[Pagallo, 2012] U. Pagallo, Cracking down on autonomy: three challenges to design in IT Law, Ethics Information Technology (14) 2012, pp. 319-328.

[Roio et al, 2017] Denis Roio, James Barritt, Jaap-Henk Hoepman, Mark de Villiers, Tom Demeyer (2017), DECODE Whitepaper (in preparation)

[Sweeney, 2002] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

Wood, 2014] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151, 2014.

[Zyskind et al., 2015] G. Zyskind and O. Nathan. "Decentralizing privacy: Using blockchain to protect personal data." Security and Privacy Workshops (SPW), 2015 IEEE. IEEE, 2015.

[Zyskind et al., 2015] G. Zyskind, O. Nathan, and A. Pentland. "Enigma: Decentralized computation platform with guaranteed privacy." arXiv preprint arXiv:1506.03471 (2015).