JeDEM 9(2): 110-133, 2017
ISSN 2075-9517
http://www.jedem.org

Tokenized Ecosystem of Personal Data — Exemplified on the Context of the Smart City

Jan T. Frecè, Thomas Selzam

Bern University of Applied Sciences, E-Government Institute, Brückenstrasse 73, 3005 Bern, Switzerland, jan.frece@bfh.ch, thomas.selzam@bfh.ch

Abstract: Data driven businesses, services, and even smart cities of tomorrow depend on access to data not only from machines, but also personal data of consumers, clients, citizens. Sustainable utilization of such data must base on legal compliancy, ethical soundness, and consent. Data subjects nowadays largely lack empowerment over utilization and monetization of their personal data. To change this, we propose a tokenized ecosystem of personal data (TokPD), combining anonymization, referencing, encryption, decentralization, and functional layering to establish a privacy preserving solution for processing of personal data. This tokenized ecosystem is a more generalized variant of the smart city ecosystem described in the preceding publication "Smart Cities of Self-Determined Data Subjects" (Frecè & Selzam 2017) with focus towards further options of decentralization. We use the example of a smart city to demonstrate, how TokPD ensures the data subjects' privacy, grants the smart city access to a high number of new data sources, and simultaneously handles the user-consent to ensure compliance with modern data protection regulation.

Keywords: Decentralized Data Storage, Data Self-Determination, Zero-Knowledge, Distributed Ledger Technology, Tokenization

Acknowledgement: A previous version of this paper focusing on the application of decentralized approaches in a smart city context was previously submitted to the CeDEM 17 conference.

1. Introduction

Data has been termed to be the „oil of the 21st century“ (Palmer, 2006). While the comparison might be misleading in certain aspects, one can say with certainty that data will be the fuel smart cities run on. To collect larger volumes and a broad variety of this “city fuel”, data sources from the public but increasingly also the private sphere play a huge role. Personal data becomes increasingly important through the exploding number of IoT devices in private households and private vehicles, the ever-present and diverse collection of mobile devices, and increasingly also connected, automated, and smart homes in general. Data collected that way can be used to improve the efficiency of a city or offer valuable services to its citizens (Batty, 2013). However,

regulations concerning privacy and personal data complicate the acquisition and analysis of personal data and will do more so in the future, as the privacy of citizens' personal data has become a hot topic over the last decade. As the prestigious international law firm Covington & Burling put it: "If big data is the new oil, then privacy is the new green." (Harris, 2014).

The new European General Data Protection Regulation (GDPR) dictates that data belongs to the data subject and must only be used in accordance with the data subject, even if the data has been created by a third-party (European Parliament, 2016). This fundamentally changes the perspective on data for any organization, person, or service processing personal data. It also changes today's and tomorrow's perspectives on the personal data pools of smart buildings, smart neighborhoods, and smart cities. Even more so as under the new GDPR any data with a relation to an individual is considered personal data, e.g. data concerning electricity usage and production, water, waste water, traffic movement, consumer habits, communication habits, utilization of buildings, etc. Smart cities can very well make use of such data, e.g. concerning people's driving behavior, electricity usage, air quality, day-to-day-routine, commuting pattern, water usage, heating conduct, shopping habits, recreational activities, etc. (Nuaimi, Neyadi, Mohamed, & Al-jaroodi, 2015). Up until now, data consumers could get user related data from wherever possible, combining it with data from countless other sources, analyzing it, as well as passing on the data or analytical results of it to third parties. Processing this kind of data without the data subject's explicit consent will be illegal after 25 May 2018, once the GDPR is in full effect. That and complementary regulations thus also affect smart cities, which will among other requirements have to deal with the data subjects' right for data rectification, to limit data processing, for portability of their data, for data erasure (right to be forgotten) and more.

Many data consumers are uncertain about how to react to the new regulation to ensure the legal compliance of their data processing. Uncertainty to this degree resulting in fear of a misstep can be paralyzing for any project relying on private and/or personal data. The approach being discussed in this paper aims to ease this uncertainty and reassure profit and non-profit data acquisition alike by offering access to relevant data in an easy, quick, secure and – above all – legal way. We propose an approach aiming at a tokenized (and decentralized – cf. section 2) ecosystem of personal data (TokPD) that reassures data consumers (such as cities offering data-based services) and data subjects (such as citizens, companies, organizations etc.) alike, and offers both sides an opportunity to take part in advancing data-based services, in a save, fair and legally compliant way.

For some data subjects it might seem an attractive perspective to outsource the processing of their personal data to commercial providers, who then take over the responsibility for maintenance, safe keeping, and – potentially – commercial exploitation through data consumers, thereby granting such third-party data consumers access to the data. Yet, as any commercial entity providing a service for personal data management is potentially under financial pressure to manage their customers' data for its own good instead of their customers' (Mortimer, 2011), we think this to be in breach of the GDPR's spirit of digital self-determination of the data subjects. The data and busi-

ness models being used in organizations like e.g. Midata[1] , Trusthub[2] , or myData[3] do not eliminate the need for data subjects to blindly trust whoever is managing their data and hope for their integrity, not to mention the additional security risks in case of a breach. In centralized setups such as these, a security breach has a stronger impact, as all data is concentrated in one location ready for pick-up. An attack on a centralized host of personal data is also more probable, since attacking one target hosting huge quantities of personal data is vastly more attractive than attacking hundreds or thousands of different targets hosting only a very limited amount of data each.

1) **Only encrypted data leaves the data layer**. This protects sensitive, personal data outside of its place of creation or intended storage from (malicious) observers, both during transport and storage.
2) **Only data stripped off any personal identifiers leaves the data layer**. This protects the data owner's identity, even in case of a data breach in the TokPD system, and even if all layers of encryption should become permeable.
3) **Data stored outside the data layer must not lead to the revealing of data.** This must hold true even if the encryption can be undone in the future, due to progresses in crypto breaking and hardware improvements.
4) **Neither the assembly layer nor the analysis layer has access to any unencrypted data, or unencrypted analysis results.** Such results are encrypted, with access cryptologically only possible for the consumer layer.

In section 2, the components of the TokPD approach are defined. Subsequently in section 3, those components are put into action by discussing their processes and their co-operation therein. Section 4 discusses the three requirements listed above and explains their conceptual fulfillment within the approach. In section 5 we further discuss potential areas of application outside of a smart city context.

This paper builds on the insights had in the preceding publication (Frecè & Selzam 2017). However, it goes beyond the application in the smart city context and describes an ecosystem built on the use of tokens instead of data. It also examines further steps of decentralization striving for the ultimate and distant goal of creating a truly intermediary-free system.

2. Architecture Overview

The methodology discussed in this paper consists of several roles and elements coordinated in such a way as to ensure a zero-knowledge approach and minimal exposure even in the case of failing cyphers or key lengths in the future.

The following section gives an overview of the integral parts of the TokPD approach and their functionality. The interaction and co-operation of these roles and elements are topic of section 3.

[1] https://www.midata.coop/
[2] https://trust-hub.com/
[3] https://mydatafi.wordpress.com/

2.1. Roles

In this paper, the term "role" is reserved for agents interacting with the TokPD system, while technical components of the system itself are considered "elements". There are four roles involved in TokPD:

Data subjects contribute to the creation of data related to themselves through interaction with data creators. Data subjects are often natural persons, but legal persons are also conceivable. A data subject's private data must be stored in a distributed data store that is equipped with a local TokPD node, in order to be manageable by the TokPD system. Data subjects are certainly welcome to set up, secure, run, and maintain such distributed data stores themselves. However, the more likely case is that most data creators will assume the responsibility of data storages for their respective data subjects.

Data creators are individuals, organization, or devices that create private data related to data subjects. They act under the authority of the respective data subject. The data creator is obliged to store the private data it creates in its own data store, or any data store designated by the data subject (e.g. if the data subject is running its own distributed data store or would like to store the data in a specific distributed data store).

Data consumers are individuals or organizations using services based on TokPD to have relevant data sets assembled and analyzed. Data consumers initiate their requests for analytics on the system's TokPD platform. They never get in contact with data subjects or data creators. Once the analysis request has been processed by the TokPD system and the results are ready, the data consumer collects those in encrypted form at the TokPD platform. Access to raw, unencrypted, or non-anonymized data is never granted to data consumers; it is in fact prevented through the architecture of TokPD.

Analytics providers offer computational power to data consumers for data analytics. Any analysis is to be performed on encrypted data sets; hence the analytics provider has to meet certain technical requirements to take part in a TokPD solution. Apart from technical restrictions, the data consumer is free to choose its preferred analytics provider. As the analytics provider exclusively handles encrypted data sets, and returns encrypted results only, it never has access to unencrypted data itself. As this setup follows the zero-knowledge approach (Goldreich & Oren, 1994; Goldwasser, Micali, & Rackoff, 1989), an organization can be in the role of the data consumer and the analytics provider at the same time, without compromising the data subjects' security or privacy. However, such an approach is not recommended, as it heavily depends on long-term protection by encryption. If, in such a situation, the encryption algorithm used to encrypt the analytic payload sent to the analytics provider fails sometime in the near or far future or the CPUs become fast enough to brute force it, the data consumer can use the data sent to the analytics provider to gain additional, though still anonymized knowledge. Therefore, it is not recommended to combine the roles of data consumer and analytics provider, though technically possible.

2.2. Elements

In addition to roles, the following elements form the technological part of a TokPD-based setup. It consists of physical hardware like servers and nodes, but also of software and protocols like smart contracts or the distributed ledger. The term "distributed ledger technology" (DLT) will be used throughout this paper to designate any type of distributed ledger supporting smart contracts. Nowadays such a system would probably be based on a blockchain. The architecture of TokPD does however not specifically rely on blockchain characteristics and is therefore compatible with other DLT implementations.

The TokPD platform is the heart or rather the coordinator of the TokPD solution. It manages all accounts for data consumers and data subjects, provides access management functionalities for them, and maintains an interface through which the data analysis requests of data consumers are initiated, processed, and delivered.

The platform analyses incoming requests for data analytics and collects the required analysis algorithms from the data consumer. It assembles encrypted data sets coming from the nodes, and forwards the data set together with the analytics algorithm to the analytics provider. After the analysis, it stores the encrypted analysis results and allows the rightful data consumer to collect those. The platform also serves as a validator node for the distributed ledger. Following the zero-knowledge approach, the TokPD platform has no access to any unencrypted personal data, and has no knowledge concerning the location of data sets.

Being the only centralized element of a TokPD system, the TokPD platform's exposure to attacks is naturally higher (cf. section 7 for a discussion regarding the potential of decentralization of TokPD). Therefore, the TokPD platform does not have access to any unencrypted information concerning data subjects, information pulled from decentralized data stores or analysis results. It merely coordinates the collaboration of all TokPD elements and offers a service platform for data consumers and data subjects. Therefore, and due to the compartmentalized architecture, the only data accessible on the TokPD platform in case of a successful attack and total breach is information concerning past analysis requests by data consumers and the reference metadata of data subjects' data sets (cf. section 4.1.4).

For example, instead of learning all the results of hourly conducted and saved air quality measurements at a data subject's house, an attacker of a TokPD platform would, even after a total breakdown of the platform, including its encryption, only learn, that the same data subject saved hourly measurements of air quality somewhere in a distributed data store connected to a TokPD node. By default, an attacker would even not be able to find out how many data sets are assigned to the data subject (for detail cf. section 3.1). The identity of the data subject, the location of the TokPD node, the data of the air quality measurements, all remain unknown to any attacker. The only way to get these is through compromising the data subject's login credentials, not by compromising the TokPD platform.

The local TokPD nodes are active on the distributed ledger, validating transmissions which have been created by the TokPD platform or other nodes, as well as adding blocks to the distribut-

ed ledger to reference data. Nodes create encrypted references to private data residing in distributed data stores. Those encrypted references are then added to the distributed ledger by the nodes. Local TokPD nodes handle the requests for data sets coming from the TokPD platform and aggregate the private data from the data stores into according and encrypted data sets. To preserve privacy conditions, all person-related data fields, or identifying data, are excluded by the distributed data store before assembling the data set that then reaches the local TokPD node.

Distributed data stores contain the private data created by data creators. Distributed data stores are usually controlled by data creators; however, a distributed data store controlled by one or several data subjects taking the role of their data creators' distributed data store is not unconceivable. Both implementations are supported by TokPD. In either case, the distributed data store lies outside of the TokPD system and is not administrated by it. Hence, the main requirement for distributed data stores is their ability to send anonymized data sets to the local nodes upon request.

Blockchain technology (Nakamoto, 2008) is used by the TokPD approach as one potential implementation of distributed ledger technology (DLT) capable of immutably storing data and decentrally executing smart contracts. The distributed ledger does not contain personal data of any data subjects, but merely encrypted references to it. It connects all local TokPD nodes and the TokPD platform. All nodes and the TokPD platform are functioning as validators, ensuring that every addition to the distributed ledger is compliant and every execution of a smart contract is done properly. Distributed ledger technology allows for pseudonymity for system users, yet transparency and traceability of transactions, while giving no single stakeholder control over the data channel. These features make blockchain currently the DLT protocol of choice allowing for self-determination for data subjects in combination with encryption and consequent referencing of data. The TokPD concept, however, is technologically neutral and therefore compatible with other distributed ledger technologies and does not explicitly rely on blockchain.

Tokenization in TokPD is achieved through publishing referencing metadata exclusively. A data subject's private or personal data is valuable and must therefore be protected, yet also be made accessible and utilizable under specific conditions. TokPD separates data and data operation by having tokens in the blockchain (or other distributed ledger) represent the data, the data type, and the data location, while the data subject's personal, private data itself remains outside the token. As the TokPD distributed data stores are addressed through cryptographic keys (cf. section 3.1. for details on data announcement), the token contains all the information needed to address the distributed data store without containing the actual location or identity of it. The latter two thus become irrelevant, meaning that data can even be transferred from one distributed data store to another, as long as its encryption key is transferred too. As a result, the first step towards an information-centric network has been made: data accessibility has become independent from data location or means of storage. As soon as the data subject decides to create a token for her or his data by defining the sharing conditions for it, the data referenced in this token can be accessed no matter where the distributed data store is currently located and how it would have to be addressed.

Smart contracts are executable applications stored in a distributed ledger (Buterin, 2014). In TokPD they have two functions:

1) Access conditions to personal data are defined by the data subjects themselves and stored within the distributed ledger as smart contracts. These smart contracts function as gatekeepers, providing partially automated access management to personal data. To further strengthen the data subjects' privacy, the access conditions of all new data are initially set so that no external access to the data is allowed. To grant third parties analytical access to its data, the data subjects must actively define access conditions for their data. Without any definition of these explicit access conditions, subjects' data remains unreachable from the TokPD system.
2) If access is granted, the smart contract signals the responsible node to export the requested data, compile it as an encrypted data set, and send that to the TokPD platform. As the reference to the node is encrypted, smart contracts do not have access to sufficient information to conclude which node they are contacting or even if they have contacted the node for another data set ever before. Smart contracts are executed on all nodes of the distributed ledger quasi-simultaneously, in order to compare the results and thereby expose manipulation attempts by individual nodes. For each announced data set, there is a separate smart contract residing in the distributed ledger, acting as a gatekeeper to the data set. Each time the access conditions for the data set are amended, the respective smart contract, responsible for access control to the data set, must be re-added to the distributed ledger. Thus, only the newest version of a smart contract is considered valid.

3. Process

The TokPD concept consists of three main processes covering the system's functions:

- Data Creation and Announcement
- Data Access Management
- Data Analysis Report

In the following sections, the process steps and goals are discussed in detail and are later used in section 4 in combination with the roles and elements to illustrate the fulfillment of the requirements discussed in section 1. For a better understanding, the example of a data subject and its mobility data is used throughout the sections.

3.1. Data Creation and Announcement

The "Data Creation and Announcement" process (cf. Figure 1) is for the most part the first-running process as it represents the originating point of any data set managed in a TokPD system. The process involves the following functionalities and roles:

Table 1: Data Creation and Announcement - Roles and Functionalities (source: author)

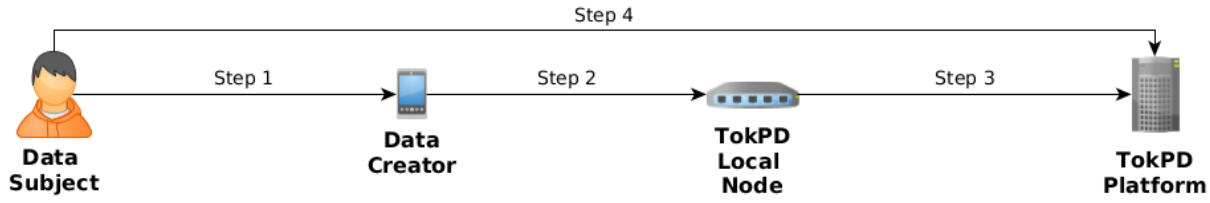
Functionalities	**Involved roles and elements**
Create data in distributed data store	Data Creator (outside TokPD)
Announce data to the TokPD system	Data Creator / Data Subject

Example: Citizens track their movements across the city by public transports, by car, by bike, and on foot using all sorts of devices. Additionally, city car plate readers, cell phone antennas, and city cameras with face recognition have additional information concerning somebody's whereabouts across a possibly extended period of time. Some citizens are willing to donate this data to the city to improve its traffic models, others consider this information too delicate to share at all, and yet others are willing to grant access to their data to commercial data consumers for a fee. The device tracking the data subject's movement thereby acts as a data creator in the data creation and announcement process.

1) In order to have the newly created mobility data announced to the TokPD system and have it mapped to the respective data subject for access management, the data subject has to grant the data creator permission (via mobile app or web app) to announce the new data set.
2) As soon as the permission has been granted, the data creator announces the data to the local TokPD node. By default, a new Wallet-ID is generated for each new data set (cf. section 4.1.3) by the TokPD platform, preventing attackers from recognizing whether one data set has the same owner as another one, just by looking at the encrypted Wallet-IDs in the distributed ledger. Alternatively, the data subject may also decide to use an already existing Wallet-ID to group data sets together for individual data overview reasons.
3) The local TokPD node creates separate encrypted references in the distributed ledger referring to the different distributed data stores the mobility data is located: at the data subject's home data repository (self-tracked mobility data), at the telecommunication company's data center (cell phone based mobility data), and in the data repository of the city security services (face recognition based mobility data).
   The only unencrypted part of the reference is the so-called Type-ID (cf. section 4.1.3), designating the type of the referenced data. In the mobility data example, the Type-ID would disclose that the referenced data is mobility data. For more detailed information concerning this step, please refer to chapter B below.

As soon as the new entries have been validated on the distributed ledger, they can be used by data consumers such as the city's planning office to refer to data without having any knowledge regarding the data's owner or its location. By default, however all access to freshly announced data is denied. Consequently, the data subject has to modify the access conditions of a data set in order to make it potentially available for analytic requests by a data consumer.

Figure 1: Data Creation and Announcement Overview (source: author)

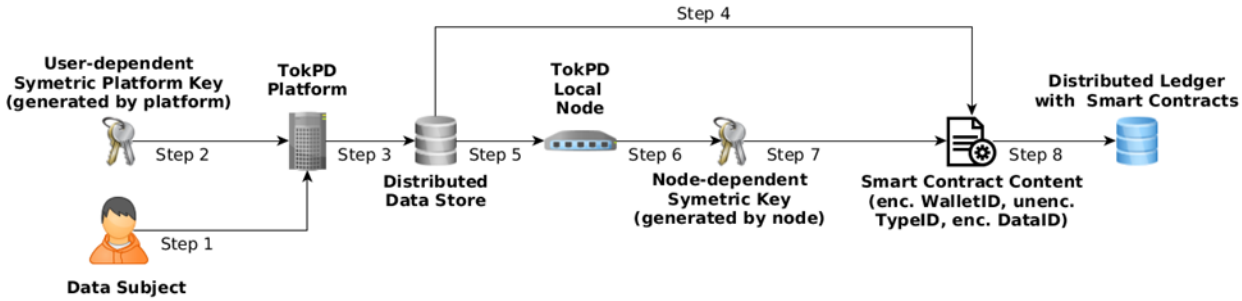


3.2. Data Announcement: Key Utilization and Smart Contract Generation

Steps 2 and 3 of the previously discussed process mention the generation of a smart contract, containing information concerning the data announced. For a better understanding of the detailed steps resulting in a smart contract (cf. Figure 2), containing an encrypted WalletID, an unencrypted TypeID, as well as an encrypted DataID are described in detail below:

1) The data subject creates a new wallet and thereby a new WalletID using the TokPD platform. While data can be added to existing WalletIDs, it is strongly recommended to use separate WalletIDs for enhanced protection.
2) A user-dependent symmetric key, created at user registration, is used by the platform to encrypt the WalletID, after having padded it with random data.
3) The encrypted WalletID is presented to the data subject. Only when the data subject decides to share this WalletID with a distributed data store, the announcement begins.
4) The distributed data store adds the unencrypted TypeID of the data to be announced, as well as the encrypted WalletID to the smart contract content.
5) The local DataID of the data to be announced is sent to the local TokPD node.
6) The local TokPD node initiates the encryption of the DataID provided.
7) A node-dependent symmetric key is used by the local TokPD node to encrypt the local DataID, before adding it to the smart contract content.
8) The local TokPD node has all the information it needs to create a new smart contract on the distributed ledger, thereby announcing the data's existence and type

Figure 2: Data Announcement: Key Utilization and Smart Contract Generation (source: author)



3.3. Data Access Management

The "Data Access Management" process can be performed upon existing, announced data sets and is exclusively triggered by the data subject. Goal of this process is an adjustment of a data set's access conditions according to the latest wishes of the data subject.

Table 2: Data Access Management - Roles and Functionalities (source: author)

Functionalities	Involved roles and elements
Adjust access conditions	Data Subject

Example: After having decided to provide its mobility data to the city, a data subject logs into the TokPD platform. There all the data sets describing the recorded movements through the city are listed. Upon being announced, the data set has been given default access conditions, prohibiting any third-party access to it. If the data subject wants to grant access to this data for analyses, it must amend the respective access conditions.

Access conditions can be formulated using all types of conditions and combinations of conditions that are verifiable by smart contracts. This may e.g. include temporal or geographical restrictions, financial compensation, restrictions bound to the total number of data sets exhibiting the same Type-ID, a counter only allowing a certain number of accesses, etc. If all defined access conditions for a data set are found to be met by a smart contract while processing a search request, the data set is considered relevant for the search and is consequently included to it. In this example, the data subject could grant access to all movements from Monday to Friday between 10 am and 4 pm older than three months in a maximal resolution of an hourly average.

3.4. Data Analysis Request

The "Data Analysis Request" process (cf. Figure 3) is triggered by a data consumer in need of an analysis of data sets fitting a certain profile. Goal of the process is not the collection of these data sets but rather the reception of an encrypted result answering the analysis question. The process is performed upon all existing and announced data sets fitting the provided profile.

Table 3: Data Analysis Request: Roles and Functionalities (source: author)

Functionalities	**Involved roles and elements**
Create and submit analysis requests	Data Consumer
Check permissions and get relevant anonymized data	TokPD platform, smart contracts, local TokPD nodes, distributed data stores
Perform analysis and return result	TokPD platform, analytics provider

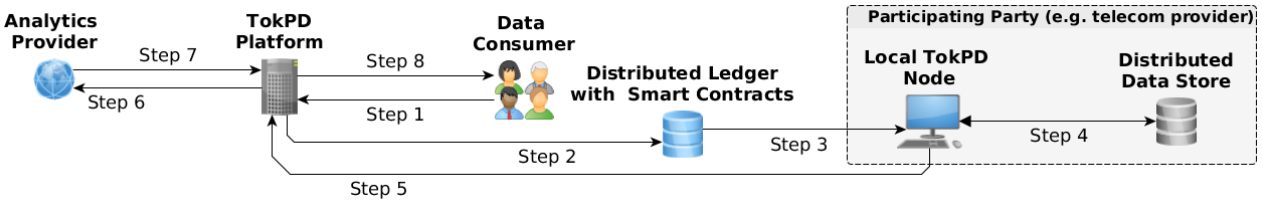
Example: A city project (data consumer) has the task to plan new public transport lines, and improve the occupancy of the existing ones. Having complete mobility data for a large number of citizens (data subjects) over months makes such a challenge much easier to tackle. In this example, the data consumer is looking for the points of the highest concentration of people trying to move through the city and their respective destinations in time slices of 15 minutes. This is the background the data consumer starts the data analysis request process against:

1) The data consumer logs in the TokPD platform and creates a new analysis request containing the analysis attributes needed and provides the respective analytics algorithm. In this case the geographic position in a 15-minute interval would be enough to determine choke points. The destination can be assumed by tracking each subject to where it comes to a longer halt.
2) The TokPD platform uses the Type-IDs of all referenced data sets to isolate the data sets fitting the data consumer's request; in this case mobility data. The TokPD platform then sends the analysis request to the smart contracts protecting the data sets fitting the analysis profile from unauthorized access.
3) Only if all access conditions defined in a smart contract are met, a data request is sent by the smart contract to the local node harboring the relevant data set. These nodes are addressed via an encrypted address pool, allowing only the recipient node to decrypt the correct address. For the access conditions described as an example in section 3.2 no data request would be sent out, as the condition of maximal one average data set per hour is not fulfilled.
4) When the affected local nodes receive the data request, they in turn request the relevant data set(s) from the local database. Only the relevant information, in this case the data subject's location in a 15-minute interval, is exported.
5) The local node encrypts the anonymized data set fully homomorphically and transfers it back to the TokPD platform.
6) The TokPD platform assembles the encrypted data sets relevant for the analysis request and finally sends the integrated and encrypted analytics payload to the data analytics provider.
7) The analytics provider receives the analytics payload and executes the algorithm, which has been provided by the data consumer in step 1 and is now forwarded by the TokPD platform, upon it. The computations can be done using either homomorphic computation (Goldwasser, Kalai, Popa, Vaikuntanathan, & Zeldovich, 2013) or secure multi-party com-

putation (Zyskind, Nathan, & Pentland, 2015). The equally encrypted result of the analysis is then returned to the TokPD platform. During this process the analytics provider does not gain any knowledge concerning the data being processed, apart from the amount of data sent for analysis. While this allows for speculations regarding the number of data sets in the respective analysis, conclusive numbers can only be calculated if the structure of a single data set and its size are known. This is prevented by the data sets' encryption.

8) When the encrypted result is sent back to the TokPD platform by the analytics provider, the data consumer is informed. He can then log in to the TokPD platform and download, decrypt, and thereby finally access the result of the ordered analysis.

Figure 3: Data Analysis Request (source: author)



Privacy and security in all processes, but especially in the data analysis request process, depend on strict role separation and seamless encryption. On the one hand this is achieved by using current ciphers and key lengths. On the other hand, it also heavily relies on key management and specialized types of encryption, as the next section emphasizes.

4. Fulfilling Privacy

In order to effectively protect the data subjects' privacy, a TokPD system has to meet the three requirements listed in the introduction section. The next sections demonstrate how two layers of anonymization and different types of encryption are applied in this concept in order to do so.

4.1. Anonymization

Anonymization has been the main tool against data abuse for decades. It is the first layer of protection against privacy violation and data abuse in the TokPD approach and is divided into two categories: data anonymization, addressing personal information in data sets, and system anonymization, focusing on filtering out any system-related information not inevitably necessary for a successful transmission.

4.1.1. System Anonymization

The address or other meta-data of an analysis request can reveal much to an attacker concerning the TokPD system itself and possible points of attack; even if the data itself remains inaccessible. In order to minimize the information exposed this way several safeguards are applied within TokPD.

4.1.1.1. Smart Contracts as Gate Keepers

In step 2 of the Data Analysis Request process, analysis requests are not sent directly from the TokPD platform to the decentralized nodes to check for relevant data, as this would reveal the decentralized nodes linked to data sets relevant for this analysis request. Additionally, such a proceeding would leak information concerning the total number of data sets fitting the Type-ID and their location.

Instead the request is sent to smart contracts saved in the decentralized ledger acting as gate-keepers keeping the identities of individual decentralized data stores unknown. By keeping this information from the TokPD platform, its exposure is heavily reduced, as the TokPD platform is the most visible part of a TokPD system and therefore the most threatened one. To counteract this exposure, the TokPD platform only works with unencrypted, accessible information (Type-IDs) to sort out the potentially relevant data sets. Furthermore, it does not directly interact with the decentralized nodes but solely communicates with smart contracts in the distributed ledger instead. This allows for the TokPD platform to lack knowledge concerning which data set is de facto relevant for the analysis and where the data is located, and nevertheless still be perfectly functional.

4.1.1.2. Decentralized Nodes Remain Hidden

To prevent illicit data drain, smart contracts only accept analysis requests signed by the TokPD platform. To prevent an identification of individual decentralized nodes, the Location-IDs are padded with random data and encrypted during their creation. This results in unique Location-IDs for each data set, only recognizable to the decentralized node administrating the related data set. When contacting the decentralized node, the smart contracts use the Location-ID as an address and broadcast the analysis request to all decentralized nodes. This setup allows the smart contracts to contact the correct node without having any information about it, apart from a padded and encrypted Location-ID.

4.1.1.3. Decentralized Data Stores Respond to their Nodes Only

Since the decentralized data store is where personal data is stored, it consequently represents the final target for any attacker. Therefore, the decentralized data stores are confined to communicating with their local, decentralized TokPD nodes only. They only accept requests signed by a known and authorized node and even the accepted requests are matched against a personal data blacklist: Data fields (attributes) included in this blacklist are not available for any kind of TokPD requests and are therefore never sent out towards the decentralized node or any other part of TokPD, even if it had been explicitly requested as a consequence of a successful attack.

4.1.1.4. TokPD Platform has Minimal Knowledge

The TokPD platform as the essential part in the middle of the TokPD system handles all incoming and outgoing requests and is therefore a high priority target for any attacker of a TokPD system. To mitigate this exposure, the TokPD platform is never handling unencrypted data or data

encrypted with a key available to the TokPD platform. Consequently, while the TokPD platform is coordinating the process from the incoming analysis request to the outgoing analysis result and therefore knows which step has to be triggered next, it has no knowledge of the TokPD nodes addressed or the results returned by them or the analysis provider. Therefore, any attacker controlling the TokPD platform would not gain any knowledge concerning personal data.

4.1.2. Data Anonymization

When a request for data arrives at the local TokPD node, the node requests the needed data from the local database. However, identity revealing data such as name, address, phone numbers, social number, physical features, religion, etc. will never be delivered by the distributed data store, even if the original request illicitly demands for it.

The risk of re-identification of individual data subjects generally depends on how much and what kind of auxiliary data an attacker can use to interrelate with the data to be re-identified. With rapidly growing volumes of publicly available data of increasing variety, the amount and quality of data that can be used to interrelate with data to be re-identified has reached a tipping-point where anonymity cannot be established anymore simply by omitting certain data fields (Narayanan, Huey, & Felten, 2016). While methods like k-anonymity or l-diversity are still used for anonymization under certain conditions, it is only a matter of time before they will become broadly useless as well (Muntés-Mulero & Nin, 2009). Years ago, Cavoukian and Castro (2014), strong defenders of anonymization approaches, warned that that "[i]n the case of high-dimensional data, additional arrangements [beyond de-identification] may need to be pursued" and the US President's Council of Advisors on Science and Technology (PCAST) adds "Anonymization remains somewhat useful as an added safeguard, but it is not robust against near-term future re-identification methods. PCAST does not see it as being a useful basis for policy." (Graham et al., 2014). Consequently, anonymization is but the first layer of protection in the TokPD approach, fulfilling the second requirement. The remaining two requirements concerning 1) the remaining of unencrypted information in the distributed data store and 2) the encryption of analysis results exclusively for the data consumer are handled via different types of encryption.

4.2. Encryption

There are two types of encryption at work in the TokPD approach. The encryption of the distributed ledger dictates how the data stored indefinitely on the ledger is encrypted. Together with the encryption of the communication between the elements of TokPD, it ensures the fulfillment of the three remaining requirements after the step of anonymization has been performed.

4.2.1. Encryption of the Distributed Ledger

Although there is no personal data stored in the distributed ledger, only indices and references, this information has to be protected nevertheless. Each entry on the distributed ledger referencing a data set consists of three elements only:

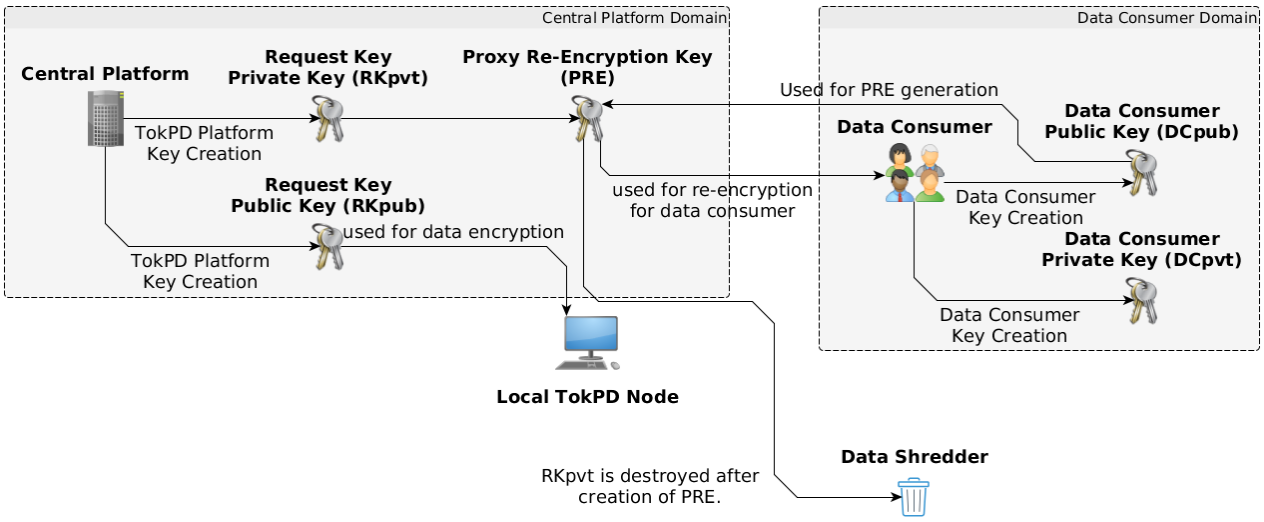
Type-ID, containing information concerning the type of data referenced. In the example used for this paper, the Type-ID would simply indicate mobility data, but it could also contain more specifics like having a separate Type-ID for cell phone-based mobility data and for face recognition-based mobility data.

Location-ID, containing the encrypted ID of the node holding the referenced data set. As the ID is padded with random data before encryption, two data set references referring to the same location would still have different encrypted Location-IDs and no relation information is leaked to an attacker.

Wallet-ID, containing the ID of the wallet used to manage this data set. The wallet's owner is automatically the owner of the data set. As each data set can be equipped with a new Wallet-ID, even data sets managed by the same data subject can be made to appear absolutely differently and no relation information is leaked.

Apart from Type-IDs, which need to be in clear text as they are used to pre-filter data sets before sending the analysis request to the smart contracts, all elements are encrypted. The data label and the Wallet-ID are encrypted using the TokPD platform symmetric key. This ensures that only the TokPD platform is able to decrypt those IDs as it is the only entity in the system with a legitimate need to know how a data creator or a data subject has decided to label a data set or in which digital wallet a data subject decided to store this data. All this information is needed to inform the data subject and enable him to take on data management himself.

Figure 4: Communication Encryption(source: author)



The Location-ID on the other hand, is encrypted using the local node symmetric key, as the local node is the only entity in the system needing to know whether a data set is stored at its own location or not. Note that nodes are only able to identify data references pointing to their own location. Locations of other nodes are unreadable to a node.

4.2.2. Communication Encryption

During the communication between data consumer, TokPD platform, local node, and analytics provider, five different keys are involved in encryption; each key responsible for a separate part of the process (cf. Figure 4).

The **data consumer public key** (DCpub) is created by the data consumer and provided to the TokPD platform for each new data analysis request. It is used by the TokPD platform in combination with the request key public key to create the proxy re-encryption key.

The **data consumer private key** (DCpvt) is created by the data consumer and kept private. It is used to decrypt the analysis results after receiving them from the TokPD platform.

The **request public key** (Rpub) is created by the TokPD platform and distributed among the local nodes. It is then used by the local nods to encrypt the data being sent to the analytics provider for encrypted computing.

The **request private key** (Rpvt) is created by the TokPD platform. It is then used in combination with the data consumer public key to create the proxy re-encryption key. After the successful creation of the proxy re-encryption key, the request private key is erased by the TokPD platform.

The **proxy-re-encryption key** (PRE) is created combining the data consumer public key and the request private key. The resulting proxy key can be used to re-encrypt data encrypted by the request public key (Rpub) in a way it can be decrypted using the data consumer private key (DCpvt). During the proxy re-encryption process the data is not decrypted and therefore not accessible.

Communication between different components of a TokPD setup is encrypted on a transport layer using TLS. Additional encryption is applied to all the data that also needs protection beyond the transportation layer.

5. Areas of Application

The ability to analyze a broad range of data sets without having access to them can be advantageous in many scenarios and in many perspectives:

5.1. Commercial Data Market

Data subjects taking part in a consumer market can hardly avoid leaving a data trail behind. The intuitive instinct to protect this data by isolating it in an offline data silo does indeed address the data and privacy protection issue. However, it does not allow for any additional value being created from this data. Some data subject might not perceive this is a problem and keep their data locked up in isolation. Others, however, might want their data to create additional value or they might be interested in selling access to their data. For both groups, TokPD offers a solution where they can easily and in detail manage access to their data, including the ability to provide access to their private data, without exposing it.

Data creators mostly do not create data for statistical reasons but for quite practical ones. A prescription is made to treat a disease, a bike is sold to replace an old one, or a flight is booked to go on vacation. None of these actions have been performed with the main intension to leave a data trail behind, but naturally they still do. This data trail potentially holds information valuable to all kinds of commercial players but nevertheless in private possession of the data subject. Therefore, every data creator must ensure the data's security and privacy. Isolating the data might solve the security and privacy problems; however, this approach would contradict the modern dogma of insight through linking and interrelating data. Hence, a system offering the data creator an easy option to store the data locally and at the same time facilitating data management directly by the data subject itself, solves many of the data creators' data privacy-related complications.

Data consumers in a commercial data market have the need for all kind of information relevant to a better understanding of their business model validity, their customers, their suppliers, their competitors, etc. Until recently, data volume and possibly data quality have been the highest obstacle on the path to new insights. With the newly introduced data protection legislation in Europe, legally compliant data handling becomes a relevant factor. Therefore, from an entrepreneur's point of view, a service offering answers based on data that is both relevant and exclusively legally compliant, definitely holds an attraction.

5.2. Public Administration

Data subjects in a public administration context are all persons being administered by the public administration, e.g. the citizens of a smart city. The data being created with reference to the data subject is only partly created on initiative of the data subject but still entirely related to it. Where the "once only principle" has not been implemented yet, data subjects, for privacy reasons, have to provide their data again and again to different administrative offices, resulting in several copies of one data set, which in turns leads to out-dated data, as soon as just one of these data set copies is not synchronously updated. In such a situation, the TokPD approach could be used to implement the "once only principle" in a decentralized way. Consequently, a data subject would profit from a tool enabling it to provide access to some parts of the data to certain individuals or institutions, without unnecessarily exposing the rest. In addition, the effort of updating data, e.g. after a relocation, is in such a system confined to the data silo hosting the data, as all other databases pull their information therefrom, resulting in a system more efficient for the administration and the citizen.

Data creators in a traditional context of public administration are all employees of a public administration that is processing data concerning private data subjects. Nowadays however, countless devices monitoring the city and its citizens produce large volumes of data: citizen-related, personal data. The role of a data creator in a commercial context closely matches the role of a data creator in an administration context. The data involved in public administration has the potential to be even more delicate or personal than in a commercial environment. The urge to isolate the data to protect it is therefore all the more pressing, and a system enabling data creators to easily store the data locally and nevertheless make the information available for other parts of the public administration in a controlled way, all the more relevant.

Data consumers in the public administration context are public services who would have been forced to get their data from the data subject directly, after getting permission for it first, and store a copy of it, time and again. This, because so far there is no suitable solution to citizens' data and let data consumers of the public administration run their queries upon it. Hence, for data consumers within the public administration a TokPD system means less duplicate data volume, less or rather no outdated copies of data, less risk of privacy violations due to manual mistakes, and a better protection of citizen data due to less exposure.

5.3. Non-human Measurements

Data subjects in a non-human measurement are hard to distinguish from data creators, as there is no direct owner of the data. Weather data measured by private citizens for a citizen-science project, air quality of a school being monitored by concerned parents, radioactivity data collected by environmental activists, all this data describes conditions per se unrelated to individual humans, such as air pressure, temperature and humidity, radioactive activity, etc. However, if this data has any relation to the individual carrying out the measurement, this individual – the data creator – in addition to the previous role becomes a data subject. This in turn means, that publishing the data including the known connection to the data subject & creator would reveal potentially personal information. Publishing them in a TokPD-system, however, would allow for a regular analysis of the data without running the risk of exposing information about the volunteers who keep the measurements going in the first place.

Data consumers for non-human measurements come in as many shapes and forms as the data consumers acting in commercial contexts, with the difference that they are not primarily looking into individual behavior but mostly measuring non-human phenomena, like the weather in a region, biodiversity in a certain area or a city's blooming patterns in hay fever season. The availability of such data, however, depends on the risk the data creators are being exposed to. It is therefore in the interest of data consumers, to use a TokPD system, as it encourages donation of data by inherently preserving the data subjects' privacy.

6. Conclusions

Implementing a TokPD approach for the data management of a smart city could have advantages on several levels and for multiple players. The citizens of a smart city with an implemented TokPD-based system could decide to support their city's data pool by providing private sensor data e.g. concerning micro-climatic conditions, traffic density, energy and water usage, air quality, energy production, etc. without risking over-sharing data by accident. Furthermore, the data subjects can be sure that only answers found through their data – and not the data itself – is presented to a data consumer. Consequently, they can also trust their data to remain where they trusted to store it.

A smart city with an implemented TokPD-based system can analyze the data shared by its citizens without risking violation of their consent by either accessing the wrong data or accessing it with the wrong intended data usage. The city can also use the same TokPD-based system to im-

plement the "once only principle" in its administration by keeping their citizens' administrative data safe in distributed data stores while having the citizens grant access to the data, if a municipal authority has a legitimate need for it.

Finally, the TokPD system can be used to facilitate access to anonymized, citizen-provided data for data consumers besides the city itself. NGOs might be interested in general traffic or environmental data, event organizers might want to collect data in the form of anonymized statistics, or groups of private enthusiasts might be interested in data concerning their passion, e.g. city microclimates or urban farming potential.

Acquiring, storing and analyzing data using a TokPD system offers data subjects and data consumers a safe, privacy-respecting, and legal way to turn personal data into fuel for smart cities. This is achieved by empowering the data subjects by giving them the option to share their personal data without having to fear for their data security and privacy, as they have to nowadays.

7. Outlook: Towards a Decentralized Platform in an Information-Centric Network

The TokPD approach, as discussed in this paper so far, achieves decentralization of most elements and functionalities through tokenization, functional separation, data distribution etc. One centralized function however, still remains: the TokPD platform, handling the user management, analysis request management, and reporting, is in the current setup unique in the entire network. Its actions cannot be validated by any peers in the network but rather have to be trusted in and accepted as the basic infrastructure's correct functioning. This setup has the usual challenges of centralized approaches: users need to trust in the system's integrity, and at the same time the platform constitutes a primary goal for crippling cyber-attacks or hack attempts aiming to shut down, block off or take over the platform, hoping to scavenge any data or just interrupt the service. Due to the setup of the system no personal data that is stored in distributed data stores would be exposed, but by taking over the platform, an attacker could still learn information concerning the different users of the system, the active data queries, and the wallet-IDs. What thus follows is a discussion on the potential, pros and cons of a decentralized TokPD approach.

7.1. Decentralizing the Approach

To achieve a decentralized system, the TokPD platform as the main intermediary within the system would have to be decentralized, as well. To realize this, the TokPD platform's functionalities must be entirely replaced by decentralized smart contracts, providing the same functionality without the vulnerability and inflexibility of a centralized approach. The TokPD platform provides four types of services to two kinds of users: data subjects and data consumers. The next chapters will point out the challenges that must be addressed to achieve a purely information-centric network of TokPD.

7.1.1. Decentralized Authentication

Authentication must be provided for all users. To avoid credentials being stored on countless foreign, decentralized devices, authentication can be performed using a private/public key pair created by the data subject or data consumer before registering for TokPD:

1) Before registration, the user's client creates a key pair consisting of a private and a public key. The public key is provided to the (decentralized) TokPD client application during registration as a unique identifier, e.g. a Secure Identity Number as used by the BitAuth (BitAuth Protocol 2015) protocol. The private key never leaves the client device, where it is stored in an encrypted file.
2) During authentication, the decentralized platform sends a sign request to the client. Normally, such a sign request would be randomly created, but since the creation process has to be replicated by all nodes of the network, it has to be set up in a deterministic way, thereby lowering the process's entropy. The client prompts the user for the password to access the locally stored and encrypted private key. After getting access to the private key, the client signs the sign request and returns it to the platform. The decentralized platform can now validate the signature using the public key provided during registration and authenticate the user.
3) As the private key never leaves the client environment and is stored in encrypted form, it can be considered a secure credential, as neither the encrypted private key nor the password used to encrypt the private key are exposed to the public at any point. To this extent, the solution even provides two-factor authentication, as the user must A) possess the encrypted private key and B) know the password to decrypt it.

As a decentralized platform, the TokPD platform also must store its authentication data in a decentralized way. Since the authentication data stored for the platform is insufficient to perform a successful authentication and the key length can be raised to a level where brute-force attacks become practically inefficient, the decentralization of the authentication part of the TokPD platform becomes implementable.

7.1.2. Access & Request Management

In a centralized setup, a user is presented with a role-dependent view after authentication. For data subjects the view consists of all their personal wallets, the data, and the access rights organized within those wallets. A data consumer gets an overview of his past and pending data requests. In a centralized setup, this information is not stored on the distributed ledger but on the centralized platform itself. Consequently, to decentralize the platform, this information would have to be decentralized, as well. However, since this information is sensitive and yet, every node in the network would have to process it to validate it, this would mean disclosing a user's data, a data consumer's request information to all involved nodes. Since storage of data and request information in a decentralized ledger is hence infeasible, a client-based composition of the information constitutes the next best solution. In this approach, the centralized platform is not decentralized using smart contracts but its functionalities are moved to the existing clients, somewhat taking the approach of a peer-to-peer network. The needed information is directly taken

by the clients from the distributed ledger and decrypted locally using the client's private keys. Either the clients act as distributed ledger nodes themselves or use a third-party service to access the distributed ledger's content. Having all the necessary keys and information locally at their disposal and access to the distributed ledger, the clients can address this task themselves.

While the data and request management functionality of the centralized TokPD platform can be moved from a centralized platform to the relevant clients, this reconstruction of the system comes at a price: All Wallet-IDs and private keys must be exclusively stored with the clients. Should they ever get lost, e.g. due to hardware failure and insufficient backups, there is no contingency; the information remains unattainable. Furthermore, security measures on private devices tend to be lower than those on professionally maintained ones. Storing crucial information on an uncontrolled, private device is therefore always an additional, highly variable risk.

7.1.3. Analysis Request

One of the critical tasks of the TokPD platform is key management during data analysis requests. From a functional point of view, the five keys in use in the context of the TokPD platform are used for encryption, decryption and proxy-re-encryption:

Encryption is exclusively done using the Request Public Key (Rpub), which is created by the TokPD platform and then distributed to the nodes to be used for encryption. In a decentralized setup, however, the key creation would not take place once, but would be performed by every node running the decentralized platform. For all the nodes to create the same key, the initial parameters for key generation must be the same for all nodes, replacing randomness with determinism as required by decentralized setups. Since keys are created in pairs or at least the private key must be known in order to create a fitting public key, all nodes running the decentralized platform would not only possess the newly created Request Public Key (Rpub) but also the fitting Request Private Key (Rpvt). Although the Request Private Key (Rpvt) is never directly used for decryption or encryption, it is used to generate the proxy-re-encryption key (PRE). It can therefore be used to decrypt the data packages sent out by the TokPD nodes. It should therefore remain private; a condition a decentralized setup can currently not fulfil.

For **decryption** the Data Consumer Private Key (DCpvt) is used. Since this key is created by the data consumer and remains with the data consumer during the entire process before being used to decrypt the downloaded analysis results, it is not affected by the decision to render the TokPD platform decentralized.

Proxy-re-encryption is performed in order to give the TokPD platform the opportunity to change the encryption performed by the nodes to an encryption the data consumer is able to access without exposing the data to the TokPD platform during the process. To create a key that is able to proxy-re-encrypt data, two keys are required: the Data Consumer Public Key (DCpub) and the Request Private Key (Rpvt). Creating the proxy-re-encryption key (PRE) in a decentralized way in turn would mean that all nodes participating in the system don't only get possession of the two keys used as an input, but also have knowledge of the resulting proxy-re-encryption key (PRE). This would jeopardize the separation of layers strived for in this setup, since all nodes would have

enough information to decrypt personal – even if anonymized – data on its way from the TokPD nodes to the TokPD platform before proxy-re-encryption.

7.2. Conclusions

We have looked into redesigning the setup from an approach where data storage is decentralized while process control remains centralized at the TokPD platform, to a fully decentralized setup. In conclusion such an approach is presently not implementable. The reason for the currently unavoidable security breach is the involvement of encryption, decryption or key creation in a decentralized setup. Not all keys used in the TokPD setup must remain secret, but the ones used to regulate the system and keep the different layers apart must. However, keeping information confidential within a smart contract setup is currently unfeasible or as expressed by Greenspan: "[F]or data hidden in smart contracts, all it takes is for someone to modify their blockchain software to display the contract's full state, and all semblance of secrecy is lost." (Greenspan 2016).

To overcome this transparency problem, encrypted computation jumps to mind. After all, the same technology is also required in the centralized setup discussed before chapter VII in order to keep the data layers isolated from each other. In the case of a decentralized setup, however, there is a crucial difference to consider. Computations based on fully homomorphic encryption are still quite inefficient and consequently very CPU-intensive. Therefore, they are best executed on resourceful hardware like mainframes or their virtualized cousins in cloud computing. Combined with the facts that A) in a decentralized setup the CPU-intensive operations would have to be performed not only once but by all the nodes of the network as a consistency check, that B) the virtual machines where smart contracts are usually executed are not as efficient as the implementations optimized for encrypted computing, and that C) the hardware in place for a distributed ledger node is usually considerably less powerful than a mainframe, the approach of implementing encrypted computing in smart contracts becomes an endeavor of the future. Without the possibility to have data processed in an encrypted form, in a decentralized way, decentralized key creation and handling are too transparent to be functional.

This leaves the question, whether redesigning a centralized control platform, with no access to data of its own, towards a more decentralized approach, where authentication as well as data and request management is addressed in a decentralized way, while encryption key creation and their application remains in a centralized form, has enough practical advantages to be propagated. While decentralized solutions generally have a high appeal, as they facilitate the removal of intermediaries, in this case the intermediary remains in the system, albeit with reduced functionality, due to authentication as well as data and request management being handled in a decentralized way, which thwarts the decentralization effort. Consequently, for the time being, solution designers have to choose between either preserving anonymity of users and data privacy by trusting a centralized intermediary handling encryption keys, isolated from any kind of readable data; or creating a decentralized solution, where no actor needs to be trusted but where options to use encryption are heavily limited, as all encryption and decryption has to be processed exclusively by clients, only using keys available to clients, and only allowed to process data owned by the specific client. There is, of course, still the possibility to again make use of encrypted computing to have

clients process (e.g. merge) data owned by other clients, but in view of the high CPU-inefficiency and the high variability of connected clients, the performance of such a setup would be volatile, at best.

Considering all the aspects of chapter VII, there are still research gaps to address and close, before data storage, access and request management, as well as authentication and data analysis can be combined into one, decentralized solution. Our discussion demonstrates once more, that distributed ledger technologies (blockchain etc.) offer great opportunities for tokenization and decentralization of ecosystems, while still facing challenges overcoming the intermediary logic, at least in contexts with greater complexity than value exchanges such as financial transactions.

References

Batty, M. (2013). Big data, smart cities and city planning. Dialogues in Human Geography, 3(3), 274–279. https://doi.org/10.1177/2043820613513390

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Etherum, (January), 1–36. Retrieved from http://buyxpr.com/build/pdfs/EthereumWhitePaper.pdf

Cavoukian, A., & Castro, D. (2014). Big Data and Innovation, Setting the Record Straight: De-identification Does Work. Information and Privacy Commissioner, 18.

European Parliament. General Data Protection Regulation (2016). Brussels. Retrieved from http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2016:119:FULL&from=EN

Goldreich, O., & Oren, Y. (1994). Definitions and properties of zero-knowledge proof systems. Journal of Cryptology, 7(1), 1–32. https://doi.org/10.1007/BF00195207

Goldwasser, S., Kalai, Y. T., Popa, R. A., Vaikuntanathan, V., & Zeldovich, N. (2013). How to Run Turing Machines on Encrypted Data. Lecture Notes in Computer Science., (8043), 536–554.

Goldwasser, S., Micali, S., & Rackoff, C. (1989). The Knowledge Complexity of Interactive Proof Systems. SIAM Journal on Computing, 18(1), 186–208. https://doi.org/10.1137/0218012

Graham, S. L., Press, W., Gates, J. S. jr, Lander, E. S., Gorenberg, M., Mundie, C., … Schmidt, E. (2014). Big Data and Privacy: A Technological Perspective. Washington DC.

Harris, M. (2014). If "Big Data Is the New Oil" Then "Privacy Is the New Green." Retrieved January 6, 2017, from https://www.insideprivacy.com/emerging-technologies/covington-at-sxsw-if-big-data-is-the-new-oil-then-privacy-is-the-new-green/

Mortimer, R. (2011, April). Could mydata be the answer to personal information problems? Marketing Week, (April 2011), 1–2.

Muntés-Mulero, V., & Nin, J. (2009). Privacy and anonymization for very large datasets. In Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09 (pp. 2117–2118). Retrieved from http://doi.acm.org/10.1145/1645953.1646333%5Cnhttp://portal.acm.org/citation.cfm?doid=1645953.1646333

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Consulted, 1–9. https://doi.org/10.1007/s10838-008-9062-0

Narayanan, A., Huey, J., & Felten, E. W. (2016). A Precautionary Approach to Big Data Privacy. Data Protection on the Move, 357–385.

Nuaimi, E. Al, Neyadi, H. Al, Mohamed, N., & Al-jaroodi, J. (2015). Applications of big data to smart cities. Journal of Internet Services and Applications, 6(25). https://doi.org/10.1186/s13174-015-0041-5

Palmer, M. (2006). Data is the New Oil. Retrieved April 8, 2016, from http://ana.blogs.com/maestros/2006/11/data_is_the_new.html

Zyskind, G., Nathan, O., & Pentland, A. S. (2015). Decentralizing privacy: Using blockchain to protect personal data. In Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015 (pp. 180–184). https://doi.org/10.1109/SPW.2015.27

About the Authors

Jan T. Frecè

Jan Frecè is an associate researcher at the BFH E-Government-Institute. Sustainable IT solutions are his main field of interest, not least due to his background of working for IBM Global Technical Services as a solution architect and a root cause analyst and his ongoing PhD in Sustainability Sciences.

Thomas Selzam

Thomas Selzam is an associate researcher and head of the "Virtual Identity" group at the BFH E-Government-Institute, tackling privacy, data protection, and ethics issues. He heads the eCH experts group IAM for Swiss e-government standardization, and is member of the Swiss Alliance for Data-Intensive Services, and the Swiss Informatics Society.

