Persistent Personal Data Vaults Empowering a Secure and Privacy Preserving Data Storage, Analysis, Sharing and Monetisation Platform

D2.2
Personal Data Market Design, Contracts and Rules

Editor(s)	Marina Da Bormida (ETA), Thanassis Giannetsos (DTU)
Lead Beneficiary	DTU
Status	Draft
Version	1.0
Due Date	30/09/2020
Delivery Date	01/03/2021
Dissemination Level	PU

DataVaults is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2019-2) under Grant Agreement No. 871755 and is contributing to the BDV-PPP of the European Commission.

Project	DataVaults – 871755
Work Package	WP2 - Security Aspects, Privacy Considerations, Value Generation and Commercialisation Outlines in Personal Data Management
Deliverable	D2.2 – Personal Data Market Design, Contracts and Rules
Editor(s)	Marina Da Bormida (ETA), Thanassis Giannetsos (DTU)
Contributor(s)	Sotiris Koussouris (SUITE5), Miguel Angel Mateo Montero (ATOS), Maria Jose Lopez Osa (TECNALIA), IFAG, Christina Tsilihkiri (OLYMPIACOS), Michail Bourmpos (PIRAEUS), Sébastien Hannay (ANDAMAN7), Ramon Ruiz (MIWENERGIA), Elena Palmisano, Paolo Boscolo (PRATO)
Reviewer(s)	Sotiris Koussouris (SUITE5)

Abstract	This deliverable elaborates on the use of Blockchain Distributed Ledgers, by DataVaults, for the creation of a digital data marketplace. This is achieved through the design and implementation of policy-compliant Blockchain structures to be enhanced with advanced on- and off-chain data and knowledge management services through the specification of appropriate security services including access control, smart contract composition (reflecting the data sharing configurations defined by the Individuals), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of trust anchors) as well as privacy-preserving services. Furthermore, detailed investigation of appropriate compensation mechanisms that can be considered in the context of DataVaults is also documented.
Disclaimer	The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein. © Copyright in this document remains vested with the DataVaults Partners

Executive Summary

The deliverable lingers over the **personal data management using smart contracts and DLTs in Datavaults**, with an in-depth analysis of SoTA approaches and focusing on the smart contract definition relevant to the project for **secure data trading**, including also the **compensation mechanisms** to be integrated.

DataVaults secure data management conceptual architecture, besides being based on the consideration of the participating entities of DataVaults and what they can do within the system, takes into account the description of the **abstract data models** considered in the project (including also the vocabulary for the data value flows), as well as a snapshot of the concrete **secure data sharing functionalities that Datavaults framework is expected to achieve through the technological building blocks and advanced cryptographic primitives.**

The overview of the conceptual architecture and workflow of actions that need to take place when a user wants to share data with other DataVaults entities, are outlined, including the data flow envisioned within DataVaults between the participating entities, combined with Smart Contracts SoTA, the initial description of the monetization services, secure trading Mechanisms, security and crypto primitives for secure data management, and smart contract composition. This sets the scene for the concrete functionalities and algorithms of DataVaults enhanced data privacy mechanisms as outlined in Section 4.

These mechanisms rely on the: i) project’s value proposition with the exact functionalities to be provided (when it comes to data privacy and anonymization) including also the conceptual architecture and workflow of actions; ii) security and trust bundles for user privacy and conveyance of data, namely **Attributed-based Encryption (ABE) for user privacy and conveyance of data, user personas and Direct Anonymous Attestation (DAA)**, as well as iii) the integration of such security and trust bundles on top of the DataVaults DLTs.

First insights of the DataVaults Compensation Mechanisms are depicted, dwelling upon:

- the economics of personal data, with reference to different approaches, to the individual’s attitude toward privacy data, to the way of giving a price to personal data, and to the Rewarding Mechanisms of Data Marketplaces;
- the monetization enabling technology, describing the current situation of Micropayments using DLT and related challenges;

On the basis of these elements, initial technology considerations for DataVaults Compensation Schemes are also drawn.

The overall purpose of this deliverable is to provide a reference document on the security and privacy-preserving trust anchors that have been selected by the consortium for integration in the overall DataVaults platform towards achieving the main vision of secure data sharing and trading services. This will be used as input to the platform’s architecture definition, the functionality of platform’s security sub-components and the further investigation, design and development of the core DataVaults security, privacy and trust bundles.

1 CONTENTS

List of Figures

List of Tables

Terms and Abbreviations

ABE	Attribute-based Encryption
DAA	Direct Anonymous Attestation
DoA	Description of the Action
SoTA	State of the Art
GDPR	General Data Protection Regulation
DFD	Data Flows Diagram
DLT	Distributed Ledger Technology
DPIA	Data Protection Impact Assessment
IoT	Internet of Things
WP	Work- Package
DTL	Distributed Ledger Technologies
BDVA	Big Data Value Association
EDPS	European Data Protection Supervisor
PCRs	Platform Configuration Registers
PID	Personal Information Diagram
PET	Privacy-enhancing Technologies
SC	Smart Contract
TPM	Trusted Platform Module

1 INTRODUCTION

The necessity of a new technology stack aiming at the provision of enhanced **data confidentiality and user privacy**, for emerging distributed data markets, is of paramount importance in cementing Europe's vision towards the realization of Next-Generation Internet and Smart Connectivity "Systems-of-Systems". In this context, the large amount of data generated and traded increases the risk to personal privacy and data security. Thus, there is an urgent need for the creation of digital data semantic marketplaces where all interested stakeholders can securely interact with each other towards leveraging and learning from the unprecedented amount of data available. Doing so will heavily contribute to the improvement of everyday lives of both citizens and businesses. But in order to materialize such enhanced data sharing, there is one crucial challenge (overarching all others) and that is **lack of trust**. Most people believe that information is a valuable commodity but is of no use if we cannot trust the source or organize it in a meaningful way.

DataVaults meets these requirements: This deliverable has a threefold objective, aimed at reporting the work and findings on the:

- Definition of the DataVaults trusted and auditable data sharing environment for a new generation of policy-compliant Blockchain structures enhanced with advanced **on- and off-chain data and knowledge management services** through the specification of novel TPM-based security and privacy-preserving protocols.
- Design of the security and privacy-preserving trust anchors that will be integrated on the selected blockchain-backed infrastructure (leveraging the Quorum technology) and that will be implemented in the context of WP3.
- First insights of the DataVaults Compensation Mechanisms that can be considered in the context of DataVaults.

The document, and the related research activities, are interrelated with most of the WPs and tasks, and in particular with:

- WP3 "Bundles for Secure Data Sharing and Access, Privacy and Trust Preservation and IPRs Management" and WP4 "Multitude Trusted Intelligence Bundles for Personal Data Insights Generation", because in them the key security and privacy-preserving bundles, presented in this deliverable, will be implemented, such as:
  - ➢ the security modules, assuring trusted and secure communication between the Personal DataVault and the DataVaults cloud-based engine, and the bundles to undertake attribute-based data asset and analytics access policies (WP3);
  - ➢ the multitude trusted intelligence bundles for searchable encryption primitives (WP4).
  - ➢ The design and development of such and other bundles will be driven by the legal and ethical requirements, as well as by the security, privacy and trust requirements, as depicted Deliverable D2.1 [13];
- WP5 "DataVaults Platform Continuous Integration", because the findings and set in this document will be reflected in the definition of the DataVaults platform architecture, as well as in the platform integration and testing;

- WP6 “Multi-Layer Demonstrators Setup, Operation and Business Value Exploration”, because it is necessary to take into account the outcomes of this deliverable in the set up and execution of the different DataVaults demonstrators’ cases, as well as, on the other side, it is important that demonstrators’ assessment and lessons learnt also cover human well-being and empowerment.

1.1 DOCUMENT STRUCTURE

The document is structured as follows:

- **Section 2** provides an overview of the data sharing and trading functionalities that need to be captured and secured by the DataVaults security and privacy-preserving trust anchors. Towards this direction, DataVaults provides enhanced **information protection and secure data management over the entire data trading process** ranging from data generation, collection and storage to data search and deletion. Within all these data operations, DataVaults integrates data security, user privacy and secure access control (Sections 3 and 4) as holistic services to allow the trusted data movement between different entities and data infrastructures.
- **Section 3** documents the design of **policy-compliant Blockchain structures** to be enhanced with advanced **on- and off-chain data and knowledge management services** through the specification of appropriate security services including access control, smart contract composition (reflecting the data sharing configurations defined by the Individuals), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of trust anchors) as well as privacy-preserving services. This way users are in control of their own privacy and that of their devices, applications and services. The data sharing relies on the abilities that will be dictated by **cryptographic trust anchors such as Attribute-Based Encryption (ABE), Access Control, and Consensus Algorithms, e.g., PoW and PoS.** The onchain information sharing is controlled by the privacy settings of the ledger, e.g., **permissioned Blockchain** – enabling only those authenticated members with the correct privileges/attributes to read, and also the implementation of a **privacy control layer via encryption**, for example, some block information is encrypted for specified member to read.
- **Section 4** documents how DataVaults achieves **data privacy** (on top of the aofrementioned security primitives) and ownership safeguarding (**privacy by design**) and **data provenance and sovereignty**. The platform uses Blockchain-based distributed ledgers for offering enhanced data and transaction security. DataVaults protects data and resources against leak or improper modifications, while at the same time ensures data availability to legitimate users. Internal storage and ledger infrastructures, handling personal and/or corporate data, can track its provenance and are regularly audited to comply with specified **security and privacy policies and regulations**. Depending on the selected privacy level, **privacy enhancement is achieved through the use of trusted computing technologies** (i.e., TPMs) as a central building block towards the provision of privacy-preserving signature schemes (Direct Anonymous Attestation (DAA). **By assuring auditable, security and privacy policy**

compliant actions, DataVaults also guarantees that application ecosystems where such policies have been technically enforced are highlighted.

- **Section 5** provides some first insights of the DataVaults Compensation Mechanisms that can be considered in the context of DataVaults.
- Finally, **Section 6** concludes the deliverable.

2 DATAVAULTS SECURE DATA MANAGEMENT CONCEPTUAL ARCHITECTURE

2.1 DATAVAULTS SECURE DATA SHARING & TRADING VALUE PROPOSITION AND SERVICES

DataVaults aims to facilitate the establishment of a **digital data semantic marketplace** that can enable enhanced **data trading functionalities** by providing **secure, trusted and auditable data sharing environments** for a new generation of policy-compliant Blockchain structures enhanced with advanced **on- and off-chain data and knowledge management services** through the specification and integration of state-of-the-art security and privacy-preserving protocols. In the following chapters of this deliverable, based on the detailed SOTA analysis conducted in the context of D2.1 [13], we will present in detail the selection of the trust anchors that have been selected in the context of DataVaults including access control, smart contract composition (reflecting the data sharing configurations defined by the Individuals), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of decentralized roots-of-trust) (Section 3) as well as privacy-preserving services such as Attribute-based Encryption (ABE), User Personas, and Direct Anonymous Attestation (DAA) (Section 4).

The vision is to enable data ownership safeguarding (**privacy by design**), data provenance and sovereignty checking and trusted consent management, while respecting prevailing GDPR legislation; a building block towards trustworthy information exchange protocols in a number of verticals (i.e., Electronic Healthcare Records, Activity Tracking, Public Sector, Finance, Intelligent Transportation Systems, etc.) where secure data sharing, processing and storage is not only expected but mandated. This capacity, jointly with the increasing demand by society and companies in privacy preserving solutions, will generate new opportunities to smoothly integrate DataVaults into the existing market services.

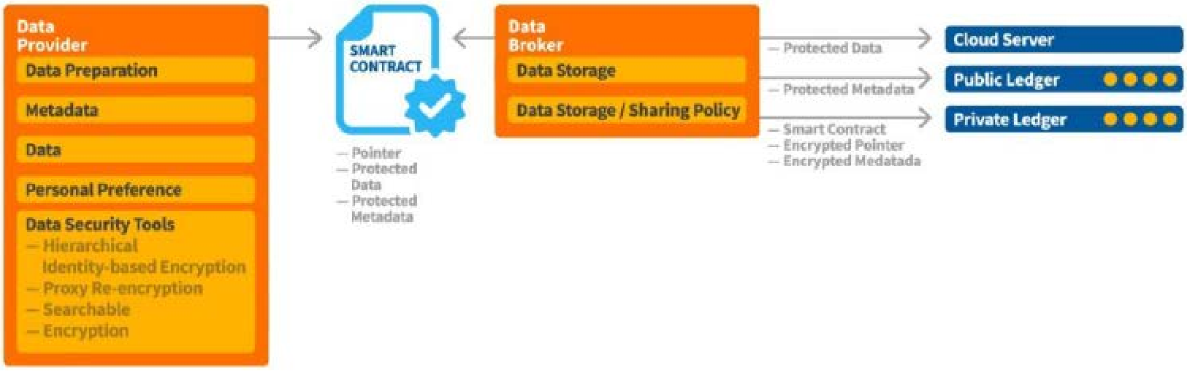


Figure 1: DataVaults Data Trading Functionalities

Towards this direction, DataVaults provides enhanced **information protection and secure data management over the entire data trading process** ranging from data generation, collection and storage to data search and deletion. Within all these data operations, DataVaults integrates data security, user privacy and secure access control (Sections 3 and 4) as holistic services to allow the trusted data movement between different entities and data infrastructures (Figure 1) leveraging a Blockchina-backed infrastructure as described in Section 4.1.

Data Preparation & Provision. This service mainly revolves around the **secure data movement** between the data provider and the DataVaults Cloud Platform (acting as the Data Broker) towards providing enhanced confidentiality. User authentication and access control mechanisms, as described in Section 3.3.1, will provide protection from data exposure. Data that is stored on distributed ledgers of the blockchain networks will also be protected. To this end, CONSENSUS will provide various **levels of data security** so that a provider is able to hide sensitive information, from network attackers or the infrastructure itself, based on his/her **personal preferences** (Section 4.2.3) and a **categorization & classification** (Section 4.2.2) based on their informational and economic value. A piece/collection of data will be segmented in two blocks: the **actual data** and the **accompanying metadata**. Metadata is formed as an abstract, structural model of the actual referenced data content to optimise visibility and searchability while providing the necessary guarantees that no sensitive information will be leaked. For instance, a collection of agricultural data in the form of "(species=watermelon, color=green, mature=80%, location=London, ……)" can be referenced by metadata with a specified category, "(metadata = "fruit data") so that it can be easily searched and indexed by a Data Seeker.

DataVaults will provide a set of **flexible and fine-grained data protection services** for managing metadata into fully plaintext, partially encrypted or fully encrypted format. This follows the principle of **user security and privacy empowerment** where users are allowed to self-determine their own security and privacy levels and that of their data depending on its sensitivity, information and economic value. They can then encrypt their data into the respective levels by using advanced cryptographic primitives such as Attribute-based Encryption (ABE) technology (Section 4.2.1). For instance, data encryption can be categorized into *extreme classified*, *classified* and *low classified* levels, so that different levels of data sharing will not affect each other's data security. Different from traditional encryption/decryption techniques, ABE allows users to maintain a unique (top level of) decryption key to control all subsequent levels of decryption guaranteeing lightweight key management. Hash values of both metadata and data are calculated as well in order to preserve data integrity.

Data Storage. This service mainly reflects the DataVaults platform feature to **securely store** the provided data in a **scalable, decentralized cloud storage market** in order to support data persistency. After receiving a data package, from the data provider, the platform stores the package into its cloud server, and further generates a pointer indexing to the corresponding storage address in the server. DataVaults offers further **fine-grained controls on metadata management** in the context of been allowed to encrypt metadata in the partially/fully encrypted format (based on the providers's data trading preference at the data market). The DataVaults platform encrypts the pointer under its identity with an unforgeable digital signature and broadcasts the package and the corresponding encrypted pointer to the provider's private channel in order to show evidence of data storage.

To promote **enhanced data sharing, marketing and trading**, different parts of the provider's data will be put on both the private and public ledgers. In the first case, the DataVaults Cloud platform fills a data trunk into a block of the private ledger including a fixed size of metadata, the corresponding encrypted pointers, transaction information, and smart contract details

(Section 3.3.2). In the second case, any authorized DataVaults component has rights to broadcast information in the public channel (data market) to form a block of the public ledger. Such a block will embed metadata and other information for enhanced searchability.

Data Collection & Searchability. In DataVaults, **secure data search and conveyance** is guaranteed as data seekers will be allowed to only search and request data of their interest (following a trading payment) without affecting the security of the remaining, uncollected data. To this end, data seekers first investigate all metadata stored in the public ledger. In the case of plaintext metadata, this processing is a straightforward task. It's for the fully/partially encrypted metadata where it gets challenging: to protect sensitive metadata from being exposed to a data collector, DataVaults offers a secure **Search Engine** [16] (integrating *searchable encryption* techniques) for offering enhanced search capabilities without revealing any core information. **This component is the premise of supporting various levels of metadata privacy.** Based on the searching results (of the public ledger), the data seeker will then identify which (private) datasets of interest he/she should trade for. The data seeker then sends a request to the DataVaults platform for agreeing on a smart contract for the data collection. Finally, the platform converts the corresponding encrypted pointers and grants access control to the cloud server (Figure 2).

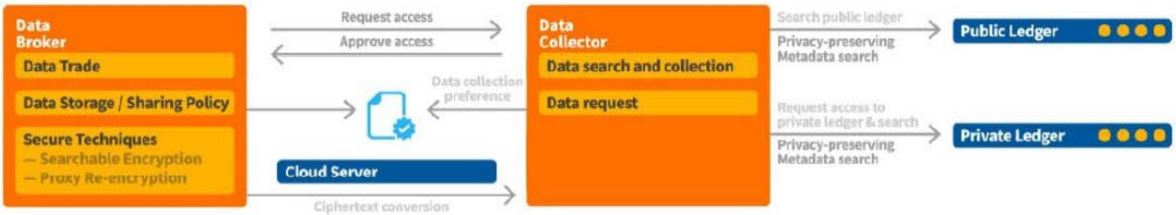


Figure 2: DataVaults Secure Data Search and Collection

Data Purge. To support enhanced user privacy and personal preference of data sharing, DataVaults offers the option of data deletion from the data cloud market. After receiving a data deletion request from the data provider, the platform will first locate and delete the corresponding data package from the cloud server so that the original storage address should be linked to an empty input (i.e. no file existing). To guarantee the **correct execution** of such a delete action and to support **validation**, the platform will decrypt the encrypted pointer and broadcast it in the private channel. Furthermore, to avoid deletion from being disclaimed or tampered, the broker will also communicate this behaviour into its private ledger by forming a block with all detailed deletion information.

The main purpose of this privacy purging is to help users meet their privacy requirements and satisfy their "*Right to be Forgotten*" which will be soon enshrined by GDPR (Art. 17). DataVaults will enhance privacy purging so that users can now purge out candidate and prospect information quickly and efficiently. This concept is clearly an important one regarding **erasure of data**. When dealing with the basic blockchain operations on persistent storage, the **immutability feature**, creates some friction. In contrast to traditional mechanisms that try to simply delete all encryption keys (something that has been identified as insufficient), DataVaults alleviates this hurdle by linking the stored reference pointer to an empty data entry (something that can be verified by all internal users) and by providing advanced cryptographic techniques towards redactable blockchain structures.

2.1.1 DataVaults Data Lifecycle

The data lifecycle of DataVaults starts with an individual that decides to collect its personal data and may go until the point where this data is shared and reused by other parties. However, the data lifecycle may end at any point of this process, and this is to be decided by the data owner, at any given time, respecting in any case the data contracts that may have been signed between a data owner and a data consumer (e.g. in case access to and usage of past data has provided unconditionally, this cannot be revoked by the user, but access to future data can be prohibited).

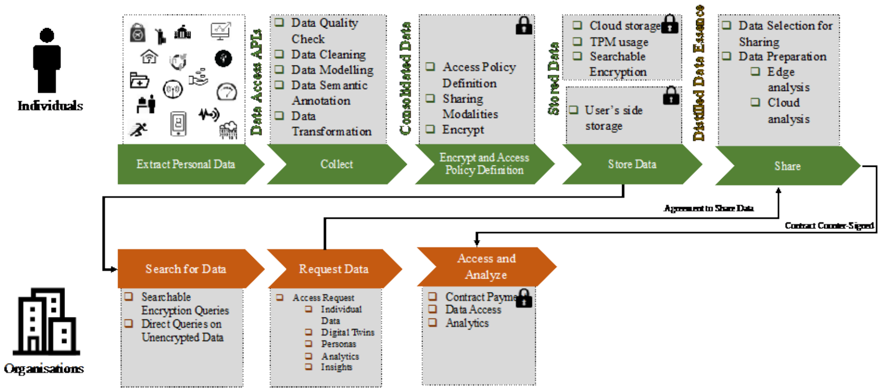


Figure 3: DataVaults Data Lifecycle

As shown in the figure above, the lifecycle of data sharing starts with data that are generated at the user side from various sensors or APIs and these are then "**Collected**" by the user. At that stage data preparation and provision activities are performed, that have to do with data quality checking, data cleaning, etc, in order to transform the data to the common schema of DataVaults. Following this, the "**Encrypt and Access Policy Definition**" step is performed, where the data provider chooses if/how to encrypt the data and decides the access policies that are to be applied on the selected data. Following this, the provider is able to "**Store the Data**" (data storage) on the DataVaults Cloud Platform. At that point, data is securely stored in the repositories and resides there until a sharing request emerges and is of course accepted. To arrive at such a situation, an external Data Seeker performs a "**Search for Data**" step (data collection and searchability), which allows it to query the data stored on the platform and identify if he would like to proceed to request it. In case that he proceeds, the "**Request Data**" step is triggered, where the data seekers define the type of data (actual data/analytics/insights) to retrieve and the nature of it (original data/digital twin data/personas data) and at that point DataVaults performs internal operations to identify, access, gather and define the value of the data necessary towards constructing a smart contract that has to be signed by the data-seeker. Upon acceptance of the contract by other parties (e.g. the data owner and the data seeker), control is moved to the platform which executes the "**Share**" stage, where the data is bundled together and released to data seeker.

2.2 DATAVAULTS USERS AND STAKEHOLDERS

Having outlined the overall DataVaults Blockchain-based architecture (Section 4.1) and the workflow of actions to enable secure data sharing and trading (Section 2.1), in what follows, we will outline who the users and stakeholders.

Data Economy stakeholders are usually broken down into **Data companies** which are organizations, public or private, that are directly involved in the production, delivery and/or usage of data in the form of digital products, services and technologies and they can be both data suppliers' and data users' organizations:

- **Data suppliers** have as their main activity the production and delivery of digital data-related products, services, and technologies and they represent the supply side of the Data Market.
- **Data Seekers and users** are organizations that generate, exploit, collect and analyze digital data intensively and use what they learn to improve their business and they represent the demand side of the Data Market and could be viewed as a major stakeholder.

But DataVaults starting position is from the viewpoint of the citizen as an individual and how that citizen shares/supplies their private data, adding to the pool available to the data economy.

- **Individual Citizen** who we define as a private person who generates and collects their own personal data from various services, devices and applications.

We further treat the mechanisms for intervening in this process as stakeholders to aid analysis. Thus we add:

- **DataVaults Personal App** which is the personal side of DataVaults, which resides at the side of Individual users.
- **DataVaults Cloud Platform** which is the central part of DataVaults architecture, residing on the cloud.
- **DataVaults Private Ledger**
- **DataVaults Public Ledger**

Finally we include:

- **DataVaults Data Scientist** who is a technical user who is familiar with big data analytics and aware of algorithms and statistics for data analysis.

The purpose of the table below is to outline the activities engaged in by the various stakeholders in order to scrutinise these discrete activities from a variety of perspectives such as: compliance with GDPR, financial accountability, in relation to value flows in the data economy, data provenance and sovereignty etc. [1]

Table 1: DataVault Data Trading Activities

Stakeholder	***What activity is being carried out in the system***	***Stage of Data Life-cycle***	***Type of smart contract***
IC Individual Citizen - adopting roles as required by the demonstrators	**IC1.** An Individual uses DataVaults Personal App to construct their unified personal data hub and collect at a single place all their personal data in a	Collection	"Private Ledger" Smart Contract (SC) with the DataVaults platform on the data sharing configuration and trading value

[1] https://ec.europa.eu/digital-single-market/en/news/building-data-economy-brochure

such as sports fans, athletes, entrepreneurs, travelers, commuters, tourists, leisure seekers, energy consumers and healthcare users.	secure and trusted manner.		
	IC2. Manages their personal data and decides what to share and with whom.	Establishing ownership of datasets (data prepation)	"Private Ledger" SC on the data sharing configurations and privacy level for the data provider
	IC3. Determination of different access levels by using appropriate security and privacy-preserving protocols – i.e., ABE, authentication, access control	Data categorization and classification	"Private Ledger" SC on the attributes and credential management for data sharing. Such SCs are also mirrored to the "Public Ledger" to be read by the data seeker
	IC4. Receives compensation for the data assets they place at the disposal of third parties.	Data trade	"Private ledger" SC for transferring the value from the platform's wallet to the wallet of the data provider
PA DataVaults Personal App Collecting personal data, configuring sharing parameters for those data, as well as analyzing them at a basic level. Visualisation	**PA1.** The DataVaults Personal App retrieves personal data from various data sources (services, devices and applications), transforms them and stores them locally.	Data preparation and collection activities. Data storage activities	No SC required
	PA2. The DataVaults Personal App offers capabilities to the Individual user to manage data access policies, configure data sharing parameters, analyze and visualize their data and remain aware of privacy exposure.	Data providers executing their rights regarding GDPR. Encrypt and Access Policy Definition.	"Private Ledger" SC on the data sharing configurations and privacy level for the data provider
	PA3. Finally, the DataVaults Personal App connects to the Private Ledger of the Individual user and interacts with the DataVaults Cloud Platform.	Data storage, sharing and purge (when requested)	"Private Ledger" SC on data management throughout the entire data lifecyle
DS Data Seeker In the demonstration phase, there are	**DS1.** Acquires Primary Personal Data from Individuals	Request Data	"Public Ledger" SC for auditable and verifiable data request. When a seeker enters the

a range of data seekers and examples of how they intend to utilise the data. They include: Municipalities and specific departments covering leisure, transport, economic development, tourism etc. Healthcare and Energy companies, Sports clubs.			DataVaults platform, and ask for some data, this SC checks if the attributes provided by the seeker match the attributes specified in the policies. This process is part of the APE (Access Policy Editor) component.
	DS2. Compensates the individual users for this data.	Compensation phase.	“Public Ledger” SC for the transfer of the required compensation from the data seeker to the platform
	DS3. Creates business intelligence based on this data.	Data Analytics	No SC required
	DS4. Combines Primary Personal Data with other types of data they already possess with a goal to create new datasets or relevant derivatives.	Data Analytics	No SC required
CP **<u>DataVaults Cloud Platform</u>** The DataVaults Cloud Platform is the central part of DataVaults architecture, residing on the cloud. A Data Seeker connects to DataVaults Cloud Platform to explore, acquire and analyze Primary Personal Data from Individuals.	**CP1.** The DataVaults Cloud Platform allows Data Seekers to search through encrypted and anonymized personal data of Individuals and express their interest to acquire them.	Search for Data Request Data.	“Public Ledger” SC for auditable and verifiable data request.
	CP2. The DataVaults Cloud Platform stores personal data of Individual users on the cloud upon user selection, indexes them and creates an Encrypted Searchable Data Lake which includes metadata and data samples and allows the operation of certain searchable encryption.	Store the Data. Data preparation activities. Data search and query	“Private Ledger” Smart Contract (SC) with the DataVaults platform on the data sharing configuration and trading value
	CP3. The DataVaults Cloud Platform anonymizes personal data of Individual	Data preparation and anonymization	“Private Ledger” SC on the data sharing configurations and

	users by implementing the Digital Twin Generator, the Persona Group Generator and by using the DAA algorithm.		privacy level for the data provider – no anonymization, partial anonymization, or unconditional anonymity
	CP4. At this point the DataVaults Cloud Platform composes and validates smart contracts, in order to grant access to the data assets for the Data Seekers and to compensate the Individual who provided them.	Compensations for data owners who let third parties use their data	"Public Ledger" SC for the transfer of the required compensation from the data seeker to the platform
	CP5. Finally, the DataVaults Cloud Platform allows Data Seekers to explore and analyze data assets and experiment within the DataVaults platform, by combining the extracts of personal data with their own (private) data and by running various analytics.	Search for Data. Access and Analyse Determine the way data can be re-used by any engaged party.	"Public Ledger" SC for auditable and verifiable data request. When a seeker enters the DataVaults platform, and ask for some data, this SC checks if the attributes provided by the seeker match the attributes specified in the policies. This process is part of the APE (Access Policy Editor) component.

2.3 DATA MODELS AND ASSETS

The DataVaults data model is defined to facilitate the interoperability and harmonisation of descriptions (metadata) of heterogeneous data to be operated by the DataVaults platform. The model is defined using the resource description framework (RDF) [1] and identifies related ontologies, concepts and vocabularies. The core DataVaults model is specified as profile of the general data catalogue vocabulary (DCAT) [2] which is an RDF vocabulary designed to facilitate interoperability between data catalogues published on the Web. Due to the open nature of RDF, the data model can be extended without breaking the system or APIs under development. At the time of writing this document, the model consists of the following parts:

- DCAT RDF classes describing the basic data profile
  - dcat:Catalog defining a structure for describing a curated collection of metadata about resources (e.g., datasets and data services in the context of a data catalogue).

  - ➢ dcat:Dataset defining a structure for describing a collection/set of data, published or curated by a single publisher, and available for access or download in one or more representations (data formats).
  - ➢ dcat:DataService defining a structure for describing a collection of operations that provides access to one or more datasets or data processing functions.
  - ➢ dcat:CatalogRecord defining a structure for describing the registration of a dataset or another resource in the catalogue. For example, when a dataset was registered in the catalogue.
  - ➢ dcat:Distribution defining a structure for describing available serializations of the dataset that may differ in various ways, including natural language, media-type or format, schematic organization, temporal and spatial resolution, etc.
- Properties for describing individuals using properties of the FOAF [3] ontology, complemented by properties from the vCard [5] ontology.
- The schema for representing access control policies in DataVaults using the ODRL Information Model [4] and Data Privacy Vocabulary (DPV) [7].
- The DataVaults data model to describe health data based on the Andaman7 AMI (Atomic Medical Item) dictionary[2].
- The DataVaults data model to describe social and activity data from social platforms like Facebook, Twitter or Instagram defined using the Semantically Interlinked Online Communities (SIOC) Core Ontology [12] and the FOAF Vocabulary [6] for edditional properties about individuals and content creators.
- The DataVaults data model to describe smart home energy data using the OEMA Energy and Equipment ontology and the OEMA Infrastructure ontology

The domains have been chosen based on the input of the project's demonstrators and the chosen properties follow the demonstrator's requirements. Deliverable D1.2 presents the data model in more detail.

[2] http://developers.andaman7.com/guide/medical-data/types.html#amis

3 DATAVAULTS SMART CONTRACTS FOR SECURE DATA SHARING

As has been described in D2.1 [13], the vision of DataVaults is to provide a **secure, trusted, auditable and privacy-preserving platform for data trading functionalities**. This is achieved through the design and implementation of **policy-compliant Blockchain structures** to be enhanced with advanced **on- and off-chain data and knowledge management services** through the specification of appropriate security services including access control, smart contract composition (reflecting the data sharing configurations defined by the Individuals), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of trust anchors) as well as privacy-preserving services.

This way users are in control of their own privacy and that of their devices, applications and services. Users are able to participate in the specification of privacy-related policies, which will then be codified in smart contracts, following the principle of **user privacy empowerment**. Depending on the selected privacy level, **privacy enhancement is achieved through the use of trusted computing technologies** (i.e., TPMs) as a central building block towards the provision of privacy-preserving signature schemes (e.g., Direct Anonymous Attestation (DAA)) (see Section 4 for more details).

In this context, DataVaults will leverage a combination of **private and public distributed ledgers** [13] (Figure 14) as the **Blockchain-powered infrastructure that will facilitate the sealing of smart contracts** on the side of the Individuals, as well as their compensation for assets that have been procured by Data Seekers. All parties will be putting information and data, as transactions (*Individuals to record any data sharing with the DataVaults Cloud Platform whereas Data Seekers will record any data trading*), and further record them on the ledgers to achieve information sharing with all nodes (i.e., Data Seekers) that will be granted access rights.

The secure data storage, publish and sharing will follow the latest trends in DLTs to rely on trust anchors of different types [14], each being important in terms of some dimension of **policy, technology, data, security, assurance and more.** DataVaults relies on a combination of advanced set of **cryptographic trust anchors towards binding entities and attributes to data subjects and data principals**, as well as to actors within the system that operate the DataVaults trust framework.

Here, how much information a data miner (i.e., Individual) can put into the block will be limited by the default block size (set on the genesis block), while the publishing of a block and data sharing relies on the abilities that will be dictated by **cryptographic trust anchors such as Attribute-Based Encryption (ABE), Access Control, and Consensus Algorithms, e.g., PoW and PoS.** The onchain information sharing is controlled by the privacy settings of the ledger, e.g., **permissioned Blockchain** – enabling only those authenticated members with the correct privileges/attributes to read, and also the implementation of a **privacy control layer via encryption**, for example, some block information is encrypted for specified member to read. **The information sharing function of Blockchain is also extended to support key management for crypto primitives following similar key hierarchies as the ones from the underlying Trusted Platform Module (TPM) acting as the root-of-trust.** Likewise, the information stored on chain could be publicly known, if they are not sensitive, like metadata, access policy, etc., so that Data Seekers can employ advanced searchable encryption techniques for identifying data of interest that they may want to acquire.

DataVaults Blockchain will mainly inherit the intrinsic functions from the **Quorum technology** (Section 4.3) to achieve the storage, publish and data sharing for all authenticated members, as well as data broker and outsiders who can first read the metadata on the public ledger before requesting access to any stored data. Different from current Blockchain functions, DataVaults will consider **secure onchain data searching so as to provide a privacy-preserving way for Data Seekers to search preferred information without leaking sensitive information of the data (on private ledger) before being granted read rights.** *The most challenging part will rely on the privacy-preserving searchability with highly search efficiency on ledger.* Another challenge DataVaults is going to address is the **insider access control layer using lightweight crypto – Attribute-Based Access Control –** to limit insider's view on others' block information, even if the insider has the membership of the private channel/ledger of an Individual. Besides, trusted ledger-based operation will be enhanced by the use of trusted hardware (through the use of Infineon's Blockchain Starter Kit [13]) in order to provide **trusted authentication for DataVaults member on ledger action, e.g., login, read, write, trusted blockchain wallet for crypto operation, key migration/management, and verification.** Unlike just using trusted hardware into the consensus algorithms, DataVaults will fully apply the trusted component into secure data sharing and management on ledgers.

In what follows, we will give a detailed overview of the DataVaults **on-chain operations regarding ledger data storage, publish and sharing through the composition and secure execution of smart contracts.** DataVaults, through its **DLT Engine** [16], will specify advanced protocol interfaces towards: (i) **Integrity and verification of block data** for guaranteeing that stored data has not been tampered with, (ii) **Mining validation** for ensuring that a block mined by a user is valid, (iii) **Consensus agreement** for allowing a majority or all network users to reach an agreement on block or ledger validation, (iv) **Membership authentication** for providing access control mechanisms (read & write privileges) to authenticated users of the ledgers, (v) **Undeniable actions commitment** for guaranteeing indisputable user operations over the ledgers, and (vi) **Customized block data security** for enabling users to put various levels of encrypted metadata onto the ledgers.

3.1 DATAVAULTS WORKFLOW FOR SECURE DATA MANAGEMENT

The general architecture of the DataVaults places the **DataVaults Cloud Platform as the "man in the middle" between Data Providers and Data Seekers.** As such, the role of the platform can be generally thought of being that of a Data Broker, who is doing the matchmaking between the transacting parties. However, as data can be stored in the DataVaults platform by the data owners without having a direct data purchasing request (for example, in the case for creating personas – see Section 4.2.2) and in order to guarantee the privacy of data owners, the overall system design is considering to utilise two distinct ledgers for recording the transactions between the **data owners and the DataVaults platform** (which we refer to as **"Private Ledger"**), and those between the **DataVaults platform and the Data Seekers** (which we refer to as **"Public Ledger"**).

As such, the architecture includes two **smart contracts (SC)** to be implemented as part of the workflow. Since the smart contracts are pieces of code, there should be a definition of the functions they will provide to the rest of the tools. The first step for that design is the description of those smart contracts and the interactions with other components in the general architecture (Figure 4).

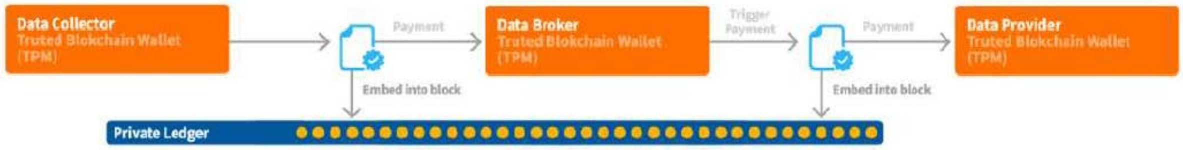


Figure 4: DataVaults Entities Data Trading

Based on the described topology, the two main processes where a smart contract is involved are the one between the Data Seekers and the DataVaults Cloud Platform and the one between the individuals (Data Owners) and the DataVaults Cloud Platform. These two smart contracts must fit all the scenarios described in deliverable D1.3 [15] and they should be synchronized in order to avoid incoherence.

In what follows, we describe in more detail the main concepts surrounding the operation and are used as the core contents of these Smart Contracts. This sets the baseline of the envisioned **functionalities of smart contracts, as a core enabler for data trading and sharing in DataVaults** that will be further modelled (including the interactions with the other DataVaults architectural components) in D5.2 [16].

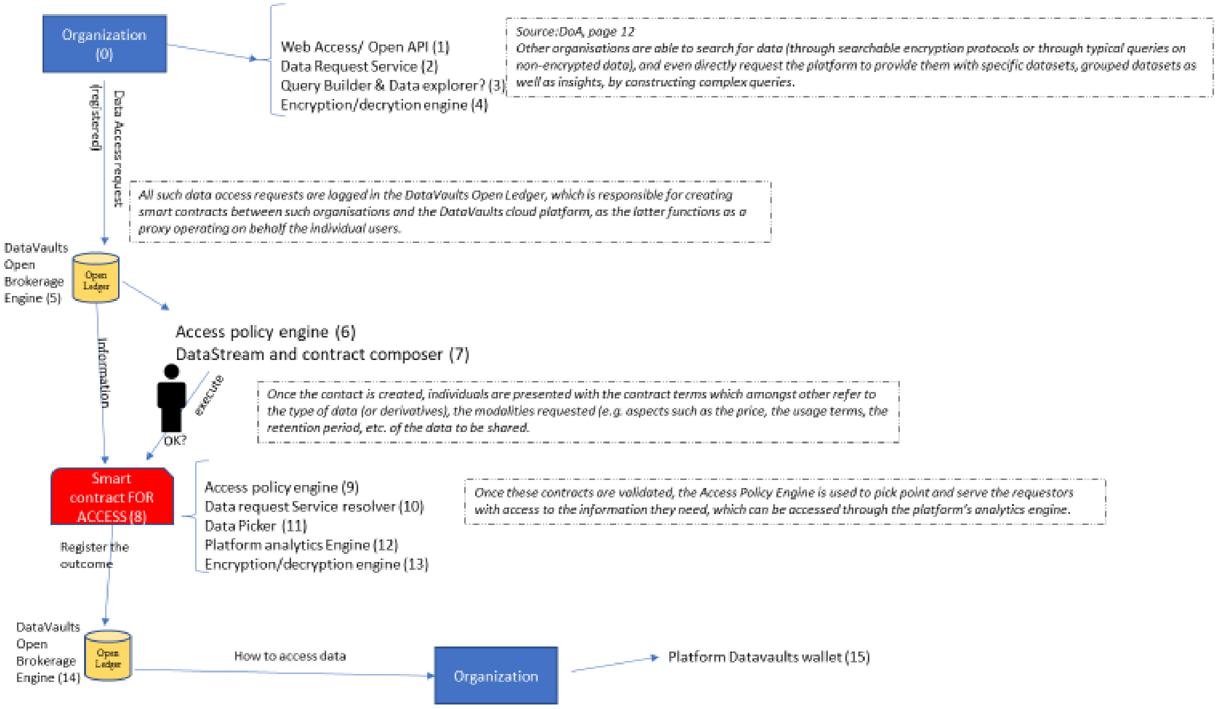
“Public Ledger” Smart Contract between Data Seekers and the DataVaults Cloud Platform:

- **Objective:** When a Data Seeker logins to the DataVaults Platform, and requests for some data, this SC checks if the attributes provided by the seeker match the attributes specified in the policies configured by the Individuals. This process is part of the APE (Access Policy Editor) component.
- **Triggered by who:** Data stream and contract composer component. It is responsible for aggregating the requests registered in the open ledger and calling the corresponding function of the smart contract.
- **The stakeholders involved:**
  - ➢ Data Seekers (external organizations) making requests and providing attributes to match to the identified policies (stored in the private ledger, of each Individual, and mirrored in the public ledger).
  - ➢ DataVaults Cloud Platform: Individuals validating the contract or the APE (Access Policy Engine) as the platform on behalf of the data providers, to check if the policies are fulfilled.
- **Input:**
  - ➢ Open ledger where the request made by the organization is registered.
  - ➢ Usage/access policies enforced by Access Policy Engine and based on the policies established by the individuals through the Access policy Editor.
  - ➢ Validations from individuals.
- **Outcome:** A transaction registered in the ledger, including the following information:
  - ➢ Data Seeker, DataVaults Platform, ID of the data. Identification of the Data seeker and the Data Owner or Owners (using a hash for not linking those with the actual data owners).
  - ➢ Type of data. It depends on the amount of heterogeneous data to be shared. It could be a list of the different types or some expression the project decides to mean the case.

- Price. The global price the Data Seeker must pay, including or not its itemization.
- Usage/access terms. The attributes of the seeker that match and allow the access and the conditions for use the data. These terms are extracted from the policies and in case they fulfil the seeker attributes, access is allowed. The terms related to the use of the data will be also reflected in this list.
- How to access the data. Once the access is allowed, the seeker needs a URL where the data are stored and instructions for it.

- **GDPR issues:** As a collection of recommendations and facts, related to the law.
  - **Controller:** Datavaults Cloud Platform
  - **Recipients:** Organizations (Data Seekers)
  - **Consent:** previously given as usage policies or through a negotiation process
  - Rights of the data subject (access, rectification, erasure, restriction, notification, data portability, object, not automated processing)

This smart contract is involved in the general workflow depicted in the following figure (Figure 5) which also depicts the Datavaults components included in each step.



Organization (0)
Web Access/ Open API (1)
Data Request Service (2)
Query Builder & Data explorer? (3)
Encryption/decrytion engine (4)
(registered)
Data Access request
DataVaults Open Brokerage Engine (5)
Open Ledger
Access policy engine (6)
DataStream and contract composer (7)
Information
execute
OK?
Smart contract FOR ACCESS (8)
Access policy engine (9)
Data request Service resolver (10)
Data Picker (11)
Platform analytics Engine (12)
Encryption/decryption engine (13)
Register the outcome
DataVaults Open Brokerage Engine (14)
Open Ledger
How to access data
Organization
Platform Datavaults wallet (15)

Source:DoA, page 12
Other organisations are able to search for data (through searchable encryption protocols or through typical queries on non-encrypted data), and even directly request the platform to provide them with specific datasets, grouped datasets as well as insights, by constructing complex queries.

All such data access requests are logged in the DataVaults Open Ledger, which is responsible for creating smart contracts between such organisations and the DataVaults cloud platform, as the latter functions as a proxy operating on behalf the individual users.

Once the contact is created, individuals are presented with the contract terms which amongst other refer to the type of data (or derivatives), the modalities requested (e.g. aspects such as the price, the usage terms, the retention period, etc. of the data to be shared.

Once these contracts are validated, the Access Policy Engine is used to pick point and serve the requestors with access to the information they need, which can be accessed through the platform's analytics engine.

Figure 5: "Public Ledger" Smart Contract schema

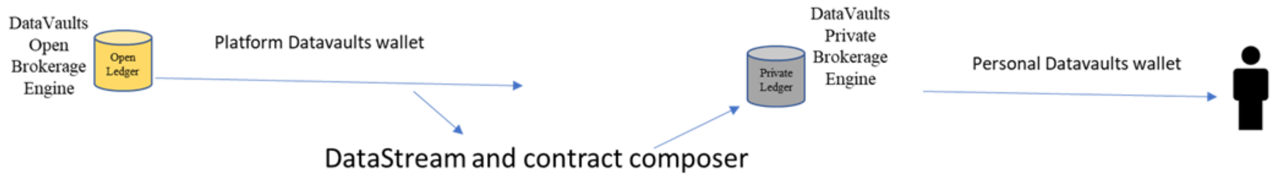
"Private Ledger" Smart Contract between Data Owners (Individuals) and the DataVaults Cloud Platform

- **Objective:** Rule and Transfer the value from the platform's wallet to the wallet of the individual user. Once the platform has cashed out the data sharing, this value (or a percentage) is transferred to the personal data platform.
- **Triggered by who:** Data stream and contract composer gathering the information and the Data Vaults DLT Engine [16] registering the transaction in the private ledger. The information about the contract between seeker and platform will be registered in the

public ledger, and the monetization mechanisms is set by the individual in its configuration and policies.

- **The stakeholders involved:**
  - DataVaults Cloud Platform: the Datavaults DLT Engine is responsible of managing the internal architecture ledgers.
  - Private and public ledgers. Providing information and registering the transaction according to the monetization mechanisms and publishing protocols.
  - Data Owners (Individuals). They are not active actors in these workflows but when the money goes to their private wallets, they will receive a message to make them aware of it.
- **Input:**
  - The information in the transaction registered in the public ledger as an outcome of the execution of the previously described smart contract, when the data was shared with the seeker.
  - The monetization preferences set by the Data Seeker (Individual).
- **Outcome:**
  - A transaction registered in the private ledger of the individual side with the information related to the payment and the data shared with who.
  - An income in the Personal DataVaults wallet of the Data Owner (Individual).
- **GDPR issues:** As a collection of recommendations and facts, related to the law
  - **Controller:** DataVaults Cloud Platform.
  - **Consent:** previously given when the individual set the policies for sharing data and the configuration of it. Automatic between platform and personal wallets.
  - Special treatment of data related to bank accounts.
  - Rights of the data subject (access, rectification, erasure, restriction, notification, data portability, object, not automated processing)

This smart contract is involved in the general workflow depicted in the following figure (Figure 6) which also depicts the Datavaults components included in each step.


DataVaults Open Brokerage Engine
Open Ledger
Platform Datavaults wallet
DataVaults Private Brokerage Engine
Private Ledger
Personal Datavaults wallet
DataStream and contract composer

Source: DoA, page 12
At the same time, the contract between the platform and the data requestor is cashed out, another contract between the platform and the individual is automatically instantiated in the Private Ledger, in order to transfer the value from the platform's wallet to the wallet of the individual user.

Figure 6: "Private Ledger" Smart Contract schema

3.2 SMART CONTRACTS STATE OF THE ART

A Smart Contract is a computer program which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement. When the Smart Contract detects the fulfilment of a preprogramed condition, it executes the corresponding action. This functionality seems very simple, but it can be extended to many complex operations to fit in numerous use-cases. An accurate definition of Smart Contract can be extracted as follows:

"A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs."

As an example, Smart Contracts are being used nowadays for:

- **Managing authorship** and implementing **pay-per-use systems** in digital works;
- **Automatic payments** for goods and services;
- Life insurance, vehicles… which payment depends on the use of the active contract;
- IoT devices and machines exchanging data for money;
- Registering the user in renting processes (houses, vehicles…).

Smart Contracts have been the impossible dream for businesses since the inception of communications and the Internet because they guarantee that every involved party has a single view of the data and prevent from fraud even when parties do not trust each other. The first generation of Smart Contracts was a traditional contract but adding a common logic for every involved party and a common verification mechanism, protected with cryptographic protocols. The most famous cryptocurrencies [17] have Smart Contracts which define the mining behaviour, transaction fees or even withdrawal limits. However, as it has been introduced before, Smart Contracts cover more use-cases than a coin exchange, ranging from financial contracts to gambling.

More recently, smart contract has been extended to other payment-related domains. Offering automation, transparency, traceability and tampered proof transactions, blockchain-based smart contracts have been becoming popular in the deployment of the sectors like government, healthcare and the real estate industry, for example, supporting quick response operations in supply chain [18], compiling the control flow and business logic, facilitating real-time order settlement of manufacturing [19], and providing hyperconnected logistics [20]. BurstIQ designs customize smart contract to set parameters of what data can be share and display of personalized health plan and status, focusing on healthcare domain. Mediachain provides smart contracts to guarantee musicians to get pay for their music copy rights and efforts. A transparency price tracing pilot is deployed in Ethereum smart contract platform [21]. Smart contracts are used in shipments, payments and track violation in supply chains [22]. Shipping industries have used smart contracts to release operations and streamline documents, e.g., [23] and [24]. In real estate marketplace, Propy develops a direct payment transaction using smart contract between buyer and seller without human interaction. Smart contract-based digital payment platforms, e.g., Circle, make use of smart contract to implement the logic of currency conversion (e.g., converting ETH to ERC20 tokens). Smart contracts can also contribute to healthcare sector by monitoring medicine selling, medical payment transactions, tracing the status of drugs [25]. Injecting data movement policy on smart contract in Ethereum [26], attribute-based access control for smart cities [27].

However, Smart Contracts are not the panacea and since they work as a computer program, they can also be hacked. It has happened several times in networks such as Ethereum, but being extensive to other Blockchains, so programmers must be cautious when developing new Smart Contracts.

As it has been stated before, Smart Contracts control the consensus mechanism, which is completely different depending on the Blockchain technology with are working with. However, unlike other technologies and systems using smart contracts only for payment, document release and policy control, in Datavaults smart contracts will be explored to woth with trusted computing technologies (Section 3.3.2), like TPMs, to certify and attest the correctness of all on- and off-chain data management operations supported through the DataVaults SCs. In this context, **DataVaults SCs aim to be among the first in the literature to merge policy control, trusted hardware execution, and remote attestation and certification.** Another featured spotlight of the design of DataVaults smart contracts is that we plan to provide interfaces between **smart contract and other crypto operations so as to provide automation and secure execution for the crypto primitives** (Section 3.3.3)**.** This is necessary for **automated data sharing and trading markets** where the use of **crypto primitives requires secure monitor, parameter, version update, and progress/execution check.** For instance, a a Data provider executes an attribute-based encryption and send the encrypted data on ledger - *how one could verify the correctness of the execution and how one could realize the update of the encryption parameter needed?* DataVaults SCs will aim to answer these questions.

Each node in the network acts as a warranty, witness and register but cannot act alone. For each transaction, each node must do some work:

- Checking that all the defined rules are followed
- If positive, creating the transaction between two involved parties
- Sharing the transaction with other nodes
- If accepted by other nodes, they will also share the transaction
- A valid state is reached for the whole network. This is the consensus.

The objective of the **consensus algorithms** is to allow the network to reach this shared state. In the DataVaults context, this is one of the most significant Blockchain operations and is mainly related with agreeing on the data sharing service between potentially **untrustworthy peers** – Data Providers and Data Seekers. **Consensus algorithms can be directly applied to block mining, transaction verification and any on-chain actions requiring the "agreement" of all/partial nodes within the network.** Instead of being seen as crypto primitive, they are more properly classified as network consensus protocols. There have been some popular consensus algorithms designed and deployed in real-world applications. Datavaults will select one or the combination of several to achieve the blockchain network consensus. Below we have brief review of them and their pros and cons.

3.2.1 Proof of Work (PoW)

Proof of Work (PoW) was the first consensus algorithm introduced with Bitcoin [17] and it is currently one of the most widely used consensus algorithms, but this is not expected to continue in the future because it is the most inefficient one as it consumes a lot of computational power. **It is required to perform a proof of work (solve a mathematical puzzle) by each one of the nodes and the winner will receive a reward.** This is what is commonly known as mining and it is a good way to prevent from spam in the network.

More precisely, PoW requires a **prover and a verifier to guess a hash puzzle and check if the puzzle is correct, respectively.** Bitcoin (SHA256), Litecoin (Scrypt), Zcash (Equihash) are the classic samples for PoW. **The pros of this mechanism are high security and decentralization. However, it also brings drawbacks on huge energy consumption** – computational resources, time consumption in finding puzzle – not applicable to efficiency-requiring network applications, and no penalty for misbehaviours. The disadvantages of PoW have moved some Blockchain applications to new types of consensus algorithms.

Furthermore, there are additional drawbacks to be considered. First of all, there is almost no incentives for mining in Bitcoin because it is not profitable anymore, due to the huge amount of computational power required to mine a block against the reduction of reward in Bitcoins. Secondly, the initial objective of this algorithm was to achieve the decentralization and democratization of the network, but in reality, there enormous mining pools which are centralizing the network because some nodes have more computational power than others inside the network.

3.2.2 Proof of Stake (poS)

In this consensus algorithm the participants with more tokens are better positioned to take decisions in the network, which is usually made by blocking some amount of cryptocurrency in a deposit. PoS [28] gets rid of the energy consumption, and no penalty shortcomings of PoW. Its core idea is to choose a block creator via various combinations of random selection based on the amount of owning currencies, called stake. And the stake will act as a guarantee that the creator will follow the correct protocol to create the block. The NXT, Nano, QTUM and Ethereum are currently making use of PoS in their platforms. A variant of PoS, called delegated PoS, was introduced in [29], which can be seen as an improvement of PoS so that nodes can select representatives through voting in order to validate blocks. It has been deployed in EOS, Cardano, TRON platforms. But in general, the weaknesses of using PoS are easily suffering from 51% attack vectors (if the number of validators is too small) and nothing at stake attack, wealth influence – richer parties may have higher influence on selection. There are also some other consensus algorithms which are quite similar to PoW and PoS requiring cost, energy and resources to do the mining, like Proof of Burn [30] (for Slimcoin), Proof of Activity [31] (e.g., Dash), Proof of Capacity [32] (for Burst).

This algorithm is actually a Proof of Work (PoW) but the amount of work that a node needs to do is proportional to the amount of cryptocurrency that he is able to block in the deposit. Also, the reward for mining the block is shared between all nodes based on their participation instead of one single node earning the whole reward, as it happened with Proof of Work.

As aforementioned, one of the advantages is its **efficiency against the PoW** and the other one is its **impartiality**, because the reward is shared between all the participant nodes.

3.2.3 Casper

Casper emerges as a **hybrid between Proof of Work and Proof of Stake** and it works as a bet where the different nodes propose the blocks to be added to the chain. The validator nodes put an amount of coins in a deposit and will receive a reward if they have been honest but will lose the deposit if they haven´t. The nodes bet for the blocks which will be added to the chain and if they guess correctly, they will receive a reward. This is the mechanism which maintains the consistency of the network.

Focusing on the security, Casper is considered as secure as the previous algorithms, but it is true that if an attacker is able to hack the betting mechanism, the system will be compromised. However, this kind of attack is almost unlinkely to succeed.

3.2.4 BFT & IBFT & RAFT

Although several pages could be written about these algorithms, only a quick introduction will be made in this section. First of all, when talking about the concept of **BFT (Byzantine Fault Tolerance), it is the ability of a distributed network to work correctly and reach consensus despite the existence of malicious nodes.** This algorithm is currently used in some private Blockchain technologies, such as Hyperledger Fabric because it provides with a mechanism to reach consensus in a network where it is supposed that a majority of the nodes is trustable.

This type of consensus algorithm [33] is based on voting process in order to add the block, and the consensus must reach when more than 2/3 of the nodes have positive vote for the block. Both PBFT and delegated variant are energy friendly, highly efficient and throughput. Ripple (ripple.com), Stellar and Zilliqa platforms are currently using this consensus type.

On the other hand, **IBFT (Istanbul Byzantine Fault Tolerance) is a variation of the POA (Proof of Authority) algorithm**, which won´t be discussed in this document. Not considering the most technical aspects of IBFT, one interesting feature of **IBFT is that it is used currently in the Quorum network and so does Raft, another consensus algorithm, being able to choose between one or another.** Similarly to BFT, these algorithms are used in private networks.

Hyperledger Fabric also supports other consensus algorithms, so it reinforces the initial idea that there are many consensus algorithms and it is not so important to know them all, rather than understand what a consensus algorithm is and what it is used for. **This type of consensus algorithms is applicable to private (permissioned) blockchain systems and non-crypto-currency applications.** There have been tremendous industrial efforts put into this direction. So far there are Fabric [34], Sawtooth [35], Burrow [36], Iroha [37] and Indy [38]. Fabric is able to provide pluggable and modular design features. It mainly uses the membership service provide scheme as identity layer to control the consensus among orderers and endorsers. Sawtooth develops proof of Elapsed Time (PoET) using the Intel SGX to build trusted execution environment for leader election. Burrow makes use of BFT to construct Tendermint consensus, while Iroha and Indy use Yet Another Consensus (YAC) and redundant BFT algorithms. Except for the PoET (lottery-like algorithm), the rest of algorithms can be used as voting. They all enjoy low energy consumption, low latency and good throughput, but also reasonable adversary tolerance – similar to PBFT algorithms.

3.2.5 Directed Acyclic Graphs (DAG)

Unlike the aforementioned consensus algorithms, **DAG is designed to use a form of data structure to make sure that information will always pass through a pre-defined direction via the existing nodes.** DAG is a blockless structure without mining process, so that it is technically faster than PoW and PoW based networks. The verification of transactions is also more efficient. Since its lightweight design, it is widely used in the IoT context. IOTA's Tangle (www.iota.org) [39] and NANO [40] are adopting the core idea of DAG in practice.

3.2.6 Considerations

In what follows, we summarize the pros and cons of the given popular consensus algorithms in Table 2. It can be seen that for designing and implementing a data trading based Blockchain platform, we may avoid using PoW and DAG. *This is so because, the former consumes huge energy resources while the latter is not in blockchain but DAG structure (although it is efficient and fast).* As for PoS, it can be supported by many prevalent blockchain platforms like Ethereum, but it requires participants to have foundation – stake – to play the consensus – being the potential leader, and it may cost a problem within a consortium – richer becomes richer, which is a bias in consensus.

Item	**PoW**	**PoS**	**PBFT**	**Hyperledger Family**	**PoWE**	**DAG**
Energy/stake consumption	High	High	Low	Low	Low	Low
Network structure	Blockchain	Blockchain	Blockchain	Blockchain	Blockchain	Graph
Decentralization	High	Middle	High	Middle	High	High
Permissionless or permissioned Preference	Permissionless	Either	Either	Either	permisisonless	Permissionless
Transaction finality	Probabilistic	Probabilistic	Immediate	Probabilistic	Probabilistic	Immediate
Mechanism	Lottery, Randomised	Probabilistic lottery, voting	Voting	Voting, lottery	voting	Voting
Scalability	Low	Medium/High	High	Medium/High	Medium/high	High
Resistance to Sybil and DoS	✓	✓	✓	✓	✓	Partial
Adversary tolerance	Less than 25% of computing power across the whole network	Less than 51% of stake of the whole network	Less than 33.3% of faulty replicas	Less than 25% of computing power	N/A	Unknown
Platforms	Bitcoin/Litecoin/Ethereum until 2018	Peercoin/Ethereum from 2018/Tendermint	XFT/Stellar/Ripple	Hyperledger Fabric	Algorand	IOTA

Table 2: Comparison among various Consensus Algorithms

From the attacking perspective, most of the consensus algorithms can hold against DoS and Sybil attacks. PoW and PoS may easily suffer from 51% attacks and double spending, while the PBFT may be vulnerable for the case when there are more than 33% of network nodes that are malicious. Besides, PoW may not be crypto-friendly because it suffers from cloud-based and web cryptojacking. Even providing less fault tolerance rate, PBFT still requires no participation cost and less than 10 seconds confirmation time (as compared to the participation cost requirement and around 100 secponds confirmation at PoW and PoS). **To enable the flexible, secure and light-weight-but-still-scalable design, DataVaults may consider combining the use of PBFT and Hyperledger family consensus algorithms (note the**

most current one is the RAFT [41]). Note the RAFT provides what PBFT does and meanwhile, it has 50% crash fault tolerance – outperforming PBFT. It is also worth mentioning that Hyperledger family can provide interface for PBFT. **DataVaults will also explore the use of Proof-of-Authority (PoA).** That is a new consensus algorithms family with high performance and fault tolerance (against 51% attacks and DoS). **In PoA, the rights of mining are given to nodes that have proven their authority, and the nodes must pass preliminary authentication.** The difficulty for DataVaults, standing at the consensus context, is to identify strong enough algorithm to support trusted hardware, crypto operations and smart contract functions, but also the algorithm still can hold against prevalent attacks on consensus.

Differently than consensus algorithms, Smart Contracts are programmable, which means that they contain the business logic implemented by a programmer, so this is the part which can be altered depending of the use case.

3.3 DATAVAULTS SECURITY & CRYPTO PRIMITIVES FOR SECURE DATA MANAGEMENT

As aforementioned, DataVaults will make use of **advanced encryption techniques** to protect user's data from being compromised and tampered by network attackers. The integration of encryption technologies will also guarantee data access rights to only authenticated and authorized system users. **Data security** includes the integrity and confidentiality of data. **User privacy** (Section 4) will be partially adhered to data security as potential security breaches of data can severely harm user privacy. On top of that, DataVaults will also consider user privacy through authentication mechanisms, privacy-preserving signatures (DAA – Section 4.2.3) and the use of smart contracts to ensure user ledger access rights, data copyright, and contract rights. **Ledger security** mainly revolves around the correct control and operation of the Blockchain structure.

Towards guaranteeing the aforementioned properties, there is a plethora of security, privacy and operational assurance algorithms and techniques that DataVaults can investigate as core building blocks in the context of secure data sharing: (i) **Target Collision-Resistance Cryptographic Hash Functions** [42], (ii) **Merkle trees** [43] where data pieces are grouped in pairs and the hash of each of these pieces is stored in the parent node. In context, a data piece is captured as one transaction record and Merkle trees are used for efficient data storage and scalability, (iii) **Searchable Encryption** [44, 45, 46] have been proposed towards enhanced security in data storage, sharing and searchability. Such models if properly designed and implemented, can enable data querying even when the data is encrypted but in a resource-efficient manner (something that has been identified as a main limitation in existing Blokchain structures), and (iv) **Digital Signatures** [47] **with various levels of anonymity** (e.g., linkable group signatures) can also be considered for achieving public verifiability and unforgeability;

DataVaults is also going to address the **insider access control layer using lightweight crypto – Attribute-Based Access Control –** to limit insider's view on others' block information, even if the insider has the membership of the private channel/ledger of an Individual. Besides, trusted ledger-based operation will be enhanced by the use of trusted hardware (through the use of Infineon's Blockchain Starter Kit [13]) in order to provide **trusted authentication for DataVaults member on ledger action, e.g., login, read, write, trusted blockchain wallet for crypto operation, key migration/management, and verification.**

3.3.1 Access Control and Data Usage Policies

Access control in DataVaults is organized around the concept of **Defense-in-Depth**. That is, instead of providing a single defense mechanism for protecting the system at its entry point, the system foresees the ability to provide several defense strategies according to the importance of the resources to be protected, so the defense is conceived in several layers, each of them with their own mitigation strategies that complement each other. This way, even if an attacker would be able to bypass one of the security mechanisms of the outer layer, it would still have to face the additional security mechanisms of the rest of layers. More specifically, the access control in Datavaults is organized around three layers: Operational, Smart Contract and Privacy as shown in Figure 7.

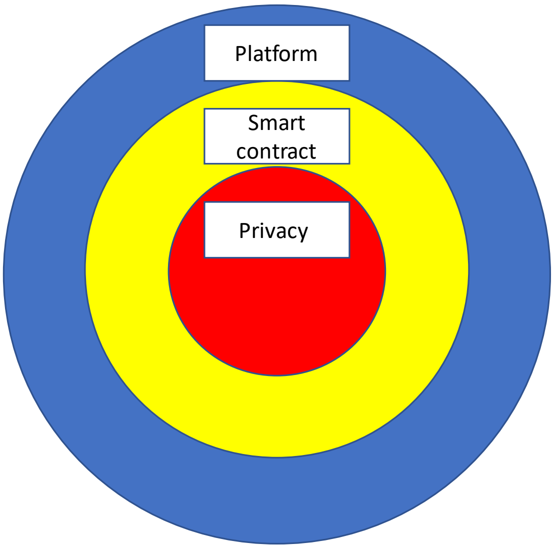


Figure 7: Access Layers in DataVaults

Platform layer will control who can access each operation in the system, regardless whether they need to make use of any of the data protected at the (private) ledger level. For instance, if a smart contract needs to be approved by a member of the commercial staff of the contractor (in order to be effective), the access operation level would control that only users that have the role of commercial staff in that company will be able to do so. More specifically, this layer provides the following functionalities:

a) Access the operation that displays all the data of a contract for DataVaults Cloud Platform to approve (which will involve the ledger level access control when retrieving several of the specific data contained in the contract.
b) Access the service if a policy is correctly checked and adhered to.

That is, as it can be seen in Figure 4 - **ledger access control complements operation access control**. Therefore, these services should be used simultaneously for different purposes.

For implementing the Platform layer, DataVaults will leverage a **Platform for Identity and Access Control (PIAM) mechanism**. **PIAMs provide authentication for users and enforce access criteria for all access points defined in them.** In the case of DataVaults, the access points to be protected will be the ones exposed by the system. According to the overall architecture (Figure 14), these points will be:

- **From the Client Application.** The Data Fetcher, the Service Resolver, and the Client Services.

- **From the Core Cloud Platform.** The Web Access.

Regarding the access criteria for these end points, the first approach of the system will be to define a **Role Based Access Control (RBAC) mechanism** to enable authenticated access to the system and the already uploaded data.

Smart Contract Layer will control that each individual datum of the smart contract can be accessed by the suitable stakeholders and is carried out by self-sovereign authentication mechanisms - as described in Section 3.3.2. For instance, it defines how only the contractor, of the smart contract, will be able to access the bank account number for emitting the charge of the contract to the corresponding bank.

Finally, the **Privacy Layer will use ABE mechanisms (Section 4.2.1) to control the access to private data.** To this end, private data, even those stored in the smart contract will be cyphered with ABE cryptography, so it will be necessary to provide a key to decipher them, making impossible for the smart layer to decrypt it without retrieving the decryption key from the legitimate user.

Attribute-based encryption (ABE), as a general extension of Public Key Encryption (PKE), is a classic type of advanced encryption, allowing sticky policies in data access control. It encrypts data under a description, so that only user(s) with the secret key matching the description can reveal the data from the encryption, in which a description could be a set of attributes or data access policies. ABE guarantees the confidentiality of data but also provide data owner policy-based data access control so that the owner can decide who can access its data via specified sticky policies.

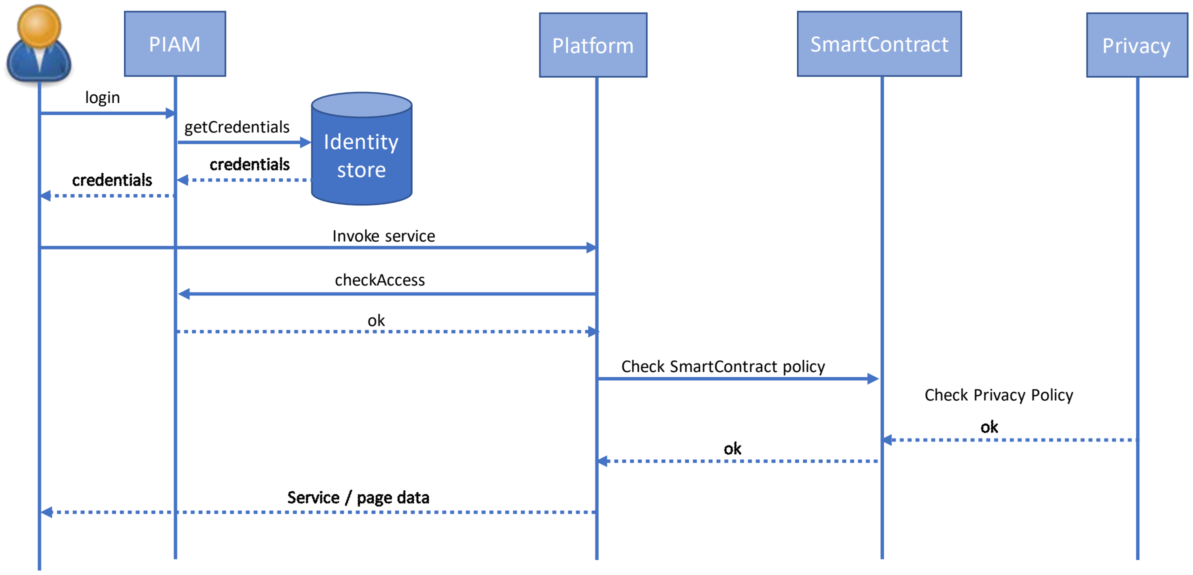


Figure 8: Access controls in DataVaults

With this mechanism, **if an attacker would get to access the page for visualizing a smart contract, it would still be clocked when trying to retrieve the data of the smart contract, resulting in a page with encrypted data.** But even if it would succeed in breaking the security of the Smart Contract Layer, it could only access non-private data because he would lack the keys to decipher the retrieved data.

Figure 8 depicts the access controls applied to the most complex case of DataVaults, which is when it is necessary to access private data shared by an Individual. As can be seen, the user needs to first authenticate in the system and the Platform layer checks with the PIAM that he/she is allowed to access the requested operation. After that, the Smart Contract layer will check the smart contract policies before accessing the data in the smart contract, and in the case of private data, it will also access the Privacy layer for enforcing the ABE mechanisms for private data.

In this context, the **Policy Management & Enforcement Toolkit** provides the capabilities of: (i) semi-automated policy creation and management based on the information that is shared by the Individual (data sharing configurations), and (ii) seamless enforcement of security and privacy policies to a set of programmable resources (including the data that can be shared by Data Seekers). The operational ecosystem of DataVaults will foster an environment where all security and privacy aspects are programmable. Spanning from hardware devices to virtualized services, there are specific APIs that can be used in order to apply policies in different layers of the DataVaults architecture. To this end, common API calls that handle the configuration of internal components such as access control, evidence handling, etc. will be formulated. Data security and privacy policies will be interpreted and enforced through the implementation of smart contracts.

The DataVaults Data Model [7] defines the lifecycle of the data within DataVaults, and it is specified as a profile of the data catalogue vocabulary DCAT [8]. *This version will be under revision for supporting the necessities that could arise during the definition of the whole system, so the part related to the scheme for representing access control to the data is a very preliminary version.*

DCAT includes in its schema the "hasPolicy" property as part of the "Resource" class, allowing to link the Resource to the ODRL information model [9] or part of it (Figure 9). Basically, **a policy is composed of rules that established the conditions to fulfill in order to set permissions or prohibitions.**

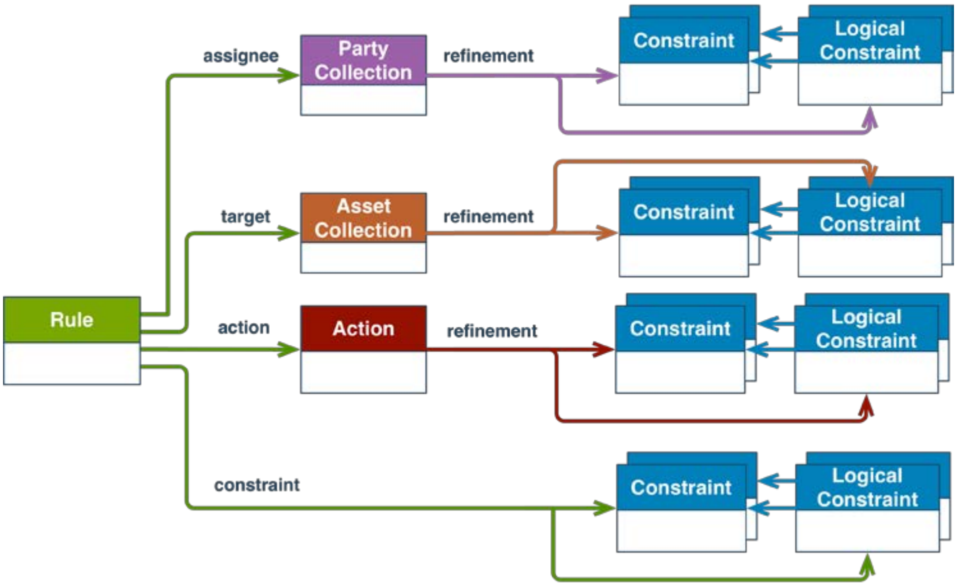


Figure 9: ODRL Constraint Relationships

The deepest level of a Rule definition shows the detailed structure for defining the conditions related to the access control, and are specified by a comparison using the constraint element. Basically the conditions set previously by the user, have to be compared to the attributes provided by the interested data seekers, and it is in this level of the policy where the access is managed.

The notation for expressing comparison involves **operators and operands.** Operators can be relational such as “greater than” or “equal to” between the left and right operands. The leftOperand property will correspond to the attribute used for this specific rule and the rightOperand will the value of the attribute that the seeker has informed.

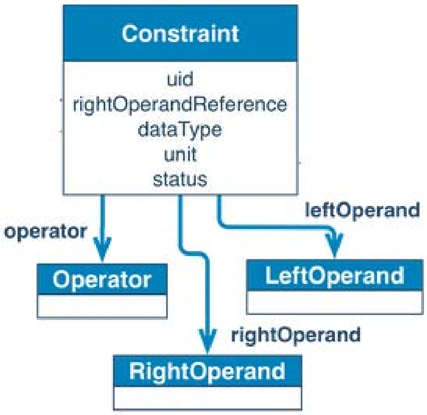


Figure 10: ODRL Constraint expression

The properties selected from de ODRL model for this comparison are two and taken from the Data Privacy Vocabulary [5]. The “hasProcessing” property is linked to the Processing categories and are related to the kind of process will be allowed, such as Analyze, Combine, Coy, Derive, Disseminate, Transform, Use, etc. The “hasPurpose” property refers to the types of purposes for which the data are going to be used. The purpose can be seen as an attribute of the Data Seekers, considered as part of the information handled by DataVaults platform related to each company and defining it.

The policy then will be built as an operation of two operands, following a notation similar to the example:

```
"permission": [{

    "action": "copy",
    "constraint": [{
      "leftOperand": "seekerType",
      "operator": "isA",
      "rightOperand": { "@dpv": "AcademicResearch" }
    }]
  }]
```

The example describes a situation when the seeker wants to copy the data and the condition for that action is that the required type of the company is AcademicResearch. The control can be more complex and include aspects as “use during a specific period of time” or “only for companies from a specific country”. The data usage control is related to a more extended data control and is not only concerned about granting permission or not but setting the conditions for using data once the access has been granted [10].

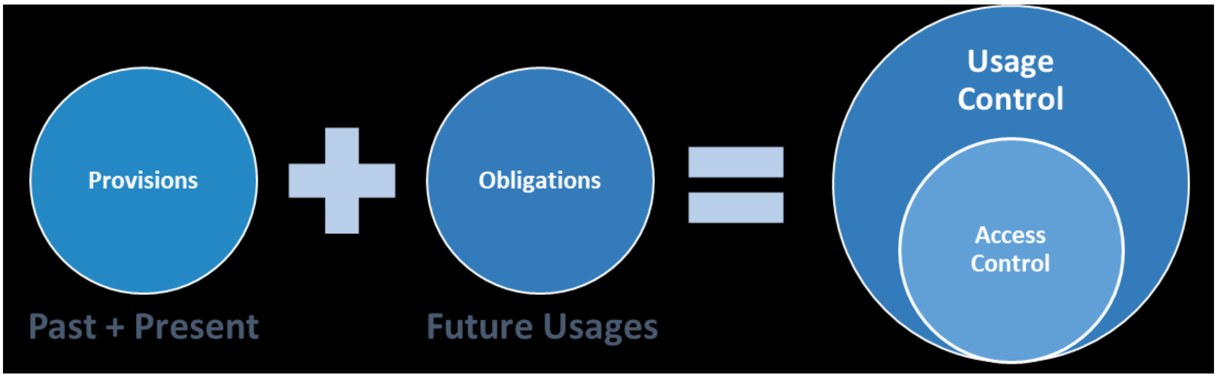


Figure 11: Usage Control – An Extension to Traditional Access Control [11]

Data usage control allows to continuously control data and prevents the incorrect use of a dataset, not following the constraints previously agreed with the data seekers and users.

At this step of the DataVaults project, there is no specific requirements about the extended control of the data, and the obligations that the companies acquire for the use of the data, are contemplated more as a "license of use" that the companies promise to comply.

In this context, the DataVaults platform considers two main modules related to the data access control - **"Policy Access Editor"** (as part of the Policy Management and Enforcement Toolkit) **and "Access Policy Engine"**.

The Policy Access editor will be a semi-graphical tool that will allow the users to define the policies for setting the conditions under which their data will be provided to the Data Seekers. It will take part of the general dashboard as a service provided by the personal DataVaults system. These policies will be stored as part of the private ledgers (mirrored also to the public ledger for being accessible to the Data Seekers) and available for the different processes and modules described in the general architecture if needed. **The Data Provider will be the unique stakeholder that could modify the policies and decide over their sharing data conditions.**

The Access Policy Engine will be an internal process, with the necessary functions available for being called from the rest of the tools and implemented mainly as smart contracts. The implementation will take as input the policies about the requested data, the attributes of the data seeker and the configuration profile defined by the user if it includes important aspects related to the data.

3.3.2 Smart Contract Computation and Verification Functionalities

Smart contracts are used as a central building block in DataVaults to manage the **access policies and privacy settings.** For this reason, all operations must be **cryptographically secured and traceable.** This section focuses on the deployment and interaction with the contract at the transaction and block level - details about the secure execution of the code are given in Section 3.4.

At this abstraction level of the system, there is no difference between **simple transactions** (e.g. sending of funds to an account) and **complex interactions with smart contracts.** Only an additional payload is attached to the transaction, which holds the contract code during deployment, or the function identifier and the parameters for function calls.

For this reason, the core security measures of the Blockchain apply for all transaction types. The following divides the combination of crypto primitives into two stages of execution: **The transaction phase covers the interaction with the end-user, where the input data is prepared and broadcasted to the network. Afterwards, the block security mechanism ensures the immutability of historic data on the Blockchain.**

3.3.2.1 Transaction security

The basic data structure for every Blockchain are individual transactions. Theoretically, every block might consist of only one transaction, but this would lead to poor efficiency and low transaction throughput. Most Blockchain technologies provide a **transaction queue to buffer incoming transactions.** The miners pick bending transactions, based on an arbitrary priority, and combine them to a block.

All Blockchain operations (e.g. transactions, contract initialisation and interaction) are encoded into transactions. **Quorum reuses the formal specification of Ethereum on this layer.** The main data fields are the recipient *address*, the transferred *value*, arbitrary *data* for contract interaction and a *nonce* to prevent replay attacks [1].

Integrity of the transaction is ensured by calculating a 256 bit *Keccak* [2] hash of the data fields. This hash is cryptographically signed by the sender using an elliptic curve digital signature algorithm (ECDSA) with the secp256k1 [3] curve. **Including the signature in the transaction guarantees that the sender is in possession of the private key.** Thus, it provides authenticity, non-repudiation and integrity of the transaction. This process is also enhanced with the integration of DAA-based signature for ensuring privacy-preservation on top of integrity (Section 4.2.3).

The signing key is the most powerful security measure at this stage: *Losing the key prevents the users from accessing their own data and funds when no backup is available*. However, backups are a thread to the security of the platform because an adversary can impersonate users when their key is exposed. This highlights the importance of secure key storage provided by TPMs or smart cards (e.g. Blockchain Security 2 Go Starterkit).

The previous steps can be performed by the user offline, without retrieving any data from the Blockchain. This circumstance can be used to simplify the implementation and further increases the usability of the solution: The application installed at the end-device (e.g. DataVaults Personal App) is not required to be a full node of the Blockchain, avoiding the need for a continuous connection to the Peer-to-peer (P2P) network. Instead, the application creates the transaction with the signature provided by the security module and submits it to a trusted entry-node (hosted by the consortium).

Further on, this allows for buffering of the transaction at the end-device when it cannot be transmitted immediately. A user could create a configuration for a new data upload on the go, and immediately sign it by authorizing a security module or using a smart card. The submission of the transaction and data upload can then be queued until a high-speed network connection is available, without the need for additional user interaction.

Every transaction is individually verified by the connected entry-nodes before they distribute it to the network. The signature is verified to confirm the data integrity and authenticate the sender. This is done to securely transfer funds from an account and optionally control permissions during execution of a smart contract. Details about secure smart contract execution are given in Section 3.4. Broadcasting of the transaction uses the default P2P propagation mechanism provided by *Ethereum*. **Combining multiple transactions into a block, the so-called *mining* or *minting* process, is dependent on the consensus algorithm and will be described in the next section.**

3.3.2.2 Block security

Only when the transaction is included in a block, it is considered accepted and cannot be easily undone. **The immutability and traceability properties of distributed ledgers are a fundamental requirement for the DataVaults platform and the main differentiator to database enabled systems.** It must not be possible, for any actor of the platform, to remove a block from the chain or to revert past transactions, or to include invalid transactions.

The consensus algorithm of the Blockchain (Section 3.2.6) dictates how new blocks are formed. In public Blockchains, the winner of the PoW puzzle can broadcast a new block, which will be accepted by the other nodes. With permissioned systems, there is usually one appointed leader responsible for the proposal of new blocks. It picks a predefined number of transactions from the waitlist, verifies the transactions with the previously mentioned methods and combines them into a block data structure.

To prevent a malfunctioning or malicious leader from appending invalid blocks, it has to go through a voting phase before being officially accepted. **Raft and IBFT in *Quorum* have a predetermined group of verifier-nodes, which use voting for electing the leader and accepting new blocks.**

Immutability of the chain is achieved by linking the blocks with cryptographic hashes of their header data. Every consecutive block contains the hash of the previous block's header data, protecting the included transactions and resulting state updates (account balances, smart contract variables) from fraudulent manipulations. For this operation the Keccak256 hash function is used again, making it infeasible for an adversary to forge a new block with a colliding hash as replacement.

The process of creating new blocks and the nodes of the network forming a consensus is described in Section 3.4.1.

3.3.3 Smart Contract Trusted Control Services

For users of the platform, **smart contracts can mostly be viewed as ordinary program code executed on some server infrastructure.** They are often used as the core logic for distributed applications (DApps) which can be seen as a new iteration of web-technologies. In the DataVaults project, **smart contracts will also play an essential role for managing the sharing preferences, access control configurations and distribution of funds.**

However, the **immutable nature of the contracts and the restrictions concerning storage and execution require additional care during design, implementation and lifecycle management.**

In contrast to public Blockchains, the **DataVaults ledgers will not support the initialization of arbitrary contract code.** The core use cases can be supported by predefined implementations which can be included in the legacy-block of the Blockchain. **This avoids the need of dynamic verification of contract code, because the official implementations are trusted and well-tested.**

The enhanced network permission model [4] of Quorum can be used to set fine-grained permissions. It is not yet clear, if this access control model gives enough flexibility for this project, or if the integration of an external provider is necessary.

In the following subsections, potential concepts for advanced smart contract management are presented. This includes solutions for accessing external data sources in smart contracts (e.g. access policies), as well as common patterns for initialization and upgrading of contract instances.

3.3.3.1 Blockchain Oracle

Smart contracts can easily access other data, which is stored on the same ledger. This includes the account balance and public attributes of other contracts where the address is known. Reading this data from external systems is also possible with simple interfaces (e.g. Web3 API for *Ethereum* [15]).

Inbound data transfer, e.g. reading off-chain data from within the smart contract to influence the program logic, is difficult. However, there are many use cases which require such input of external data. Popular examples are stock-market prices or exchange rates between different crypto currencies.

The DataVaults ledger might require input from an external policy service to update the access-policy attributes. The contract design could also implicitly query the cloud platform and confirm that the correct data was uploaded before validating the contract.

Extending the Blockchain technology with this feature can be achieved with the so-called ***Oracle*-pattern, without breaking the requirements for distributed consensus and immutability.** Retrieving data from an oracle follows a three-step procedure:

1. A user (or the platform) calls a function of the smart contract. If the needed data from an oracle is not yet stored in the contract, a request to a special oracle contract is made. An identifier for the requested data and a callback function are stored in the oracle contract. Optionally, an event is emitted to notify the external service about the request.
2. The oracle service collects the data and creates a Blockchain transaction. By calling a function of the oracle contract, the data is stored in the ledger and forwarded to the originating contract with the callback function.
3. The user is notified with an event about the data update. It can be used in decisions of the contract logic for subsequent function calls.

Alternatively, the oracle service might be authorized to change a value as soon as some external condition is met (e.g. stock-market price reaching threshold). The last step might be delayed indefinitely for this case when the condition is never observed.

One obvious disadvantage of this scheme is the added latency. Instead of a single function call, three asymmetric calls from two different parties are required to get the result. Additionally, the oracle service must be fully trusted to provide correct values with a high availability.

The oracle service also must keep a timestamped archive for all non-constant data provided. This is needed for consistently returning the same data, enabling verification of old transactions at a later time and keeping the ledger consistent.

Further investigations will decide about the need of using oracles in the DataVaults project. Because of the added complexity and trust requirement, their application must be carefully assessed.

3.3.3.2 Contract factory/minimal proxy pattern

A basic contract is initialized with a special type of Blockchain transaction. For Ethereum, this transaction contains the initialization code for preparing the persistent memory and the actual contract code as payload.

DataVaults requires a more advanced mechanism because of the following constraints:

1. Only a limited set of contracts will be permitted by the platform. Re-deploying the same contract logic frequently (e.g. upload of a new asset type by the user) causes unnecessary data duplication and accelerates the growth of the Blockchain.
2. End-users should not be allowed to deploy new contracts. However, this constraint would require an additional off-chain interface to request construction of a new contract from the platform.

There are reference implementations available to solve both constraints cleanly and exclusively within contract code.

Firstly, with the support of **Minimal Proxy Contracts** [5], an individual user does not own a full instance of the contract. Only the storage of persistent data is done on a per-user basis, the program logic is publicly stored and referenced by many contracts concurrently.

Secondly, **contracts can directly create other contracts.** This, so called **factory pattern**, is already often employed in classical software development. **In the Blockchain setting, it removes the need of special deployment operations and moves the critical initialization code to the inside of the legacy block.**

3.3.3.3 Contract upgrade

The immutability of Blockchain data, and consequently smart contracts, is a unique feature providing implicit transaction transparency and traceability. However, this prevents any subsequent changes of the implemented logic, which is not a favourable feature for developing a complex software project. Even if the probability of implementation issues can be minimized by extensive verification, any change of the target use case and new features require substantial effort because a complex re-deployment is required.

The previously introduced proxy pattern can also be used to solve this issue. **The program logic is stored in the legacy block and referenced from the individual contract instances with the unique address.** All function calls to the contract instances are forwarded to this centrally deployed logic. However, the values for the attributes are unique for every instance.

The contract factory can provide a function to allow an authorized account to change the address of the implementation. All subsequently created contracts would then reference the new contract, but already deployed contracts would keep the original logic. Figure 12 illustrates the resulting architecture.

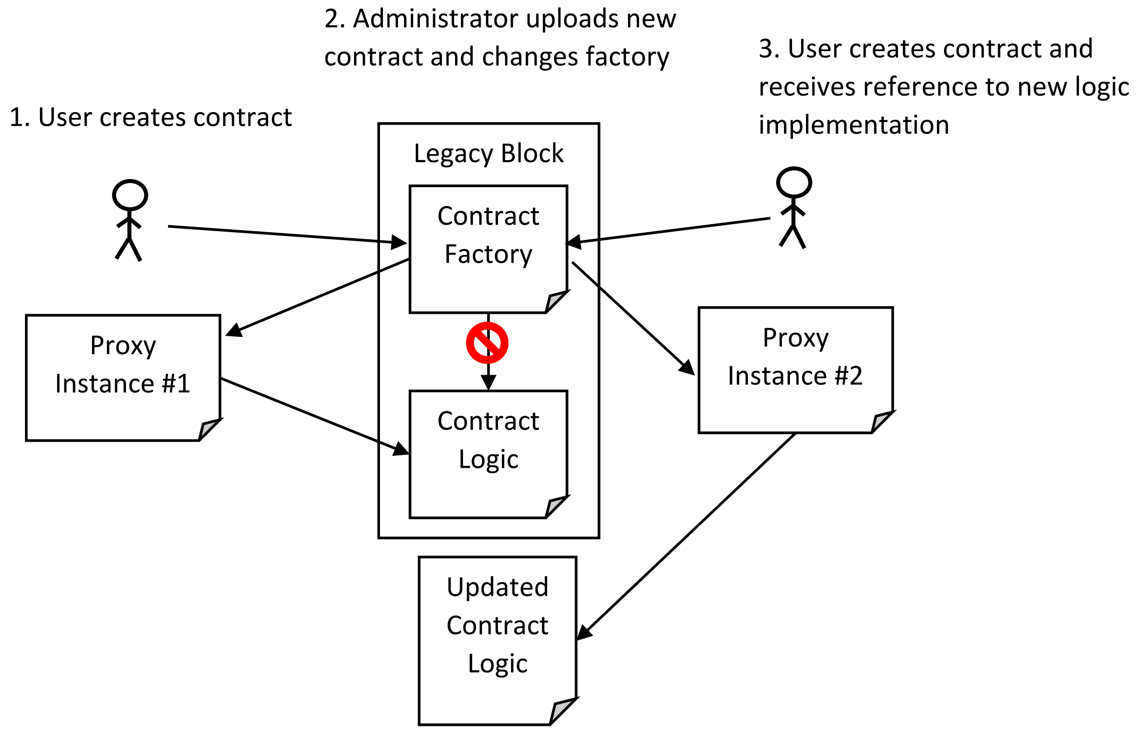

2. Administrator uploads new contract and changes factory
3. User creates contract and receives reference to new logic implementation
1. User creates contract
Legacy Block
Contract Factory
Contract Logic
Proxy Instance #1
Proxy Instance #2
Updated Contract Logic

Figure 12: Upgradable Proxy Factory

More advanced schemes can be used to even upgrade already deployed contracts dynamically. A naïve implementation would loop over all contracts and explicitly modify the referenced address. **By using a second layer of proxy delegation, the same functionality can be achieved by changing the address in a central location without modifying a large number of user-contracts.** The nested proxy is referenced by all contract instances and only stores a further reference to the currently used implementation. An administrator could then re-route all present and future contracts by changing this single reference value to an updated version. A basic overview of the main idea is illustrated in Figure 10.

The application of the previously described mechanisms will be further investigated in the next deliverables. One of the main aspect for the decision will be the expected lifetime of instances. If long-lasting contracts are expected, the nested proxy is preferred because it allows subsequent upgrades.

It is important to highlight, that this upgrade functionality does not violate the traceability properties of the Blockchain. All changes performed by an administrator are still transparently stored in Blockchain transactions. Previous contract calls can also be uniquely linked to the corresponding implementation.

Nevertheless, selection of the applied upgrade mechanism is a sensitive topic. Even if all operations are traceable, performing arbitrary modifications to the logic of deployed contracts might overstrain the trust of the users. On the other hand, it is the most effective precaution to reduce the impact of implementation issues and keep the architecture open for future extensions.

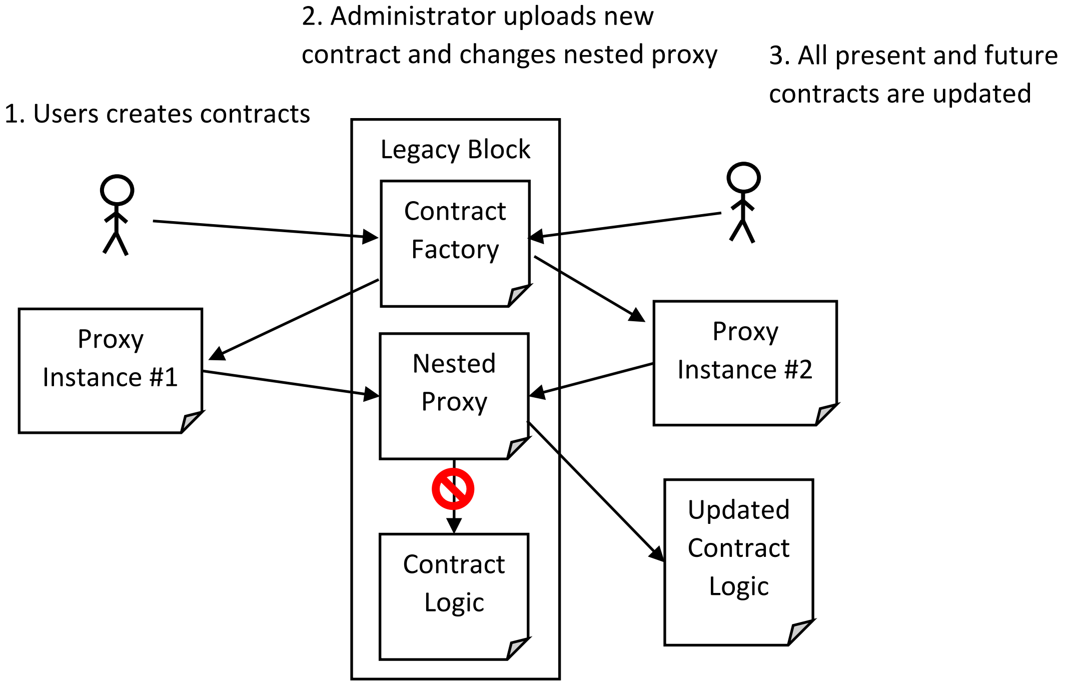

2. Administrator uploads new contract and changes nested proxy
3. All present and future contracts are updated
1. Users creates contracts
Legacy Block
Contract Factory
Proxy Instance #1
Nested Proxy
Proxy Instance #2
Contract Logic
Updated Contract Logic

Figure 2: Nested proxy contract

3.4 DATAVAULTS SMART CONTRACTS SECURE EXECUTION & OFF-CHAIN RELATED OPERATIONS

<u>**Smart Contracts Execution:**</u> Smart contract code for Ethereum and Quorum consists of instructions for the Ethereum Virtual Machine (EVM). **It is a Turing complete instruction set which is executed in an isolated sandbox by the nodes of the network.** Together with the data from the Blockchain, the contract code acts as a transaction based state machine. In this context, the term ***world-state*** **describes the common persistent data for every account and contract in the Blockchain history.** This includes the associated value (crypto currency balance) and the variable data contained in every contract.

The world-state is shared between all nodes on the network and is updated by the transactions of a block. It can be deterministically reconstructed by sequentially executing all transactions and updating the data according to the contract logic. **For efficient verification and comparison, it is stored in a Modified Merkle Patricia Tree** [1]. Changes to data stored in the leaf nodes propagates to the root node and changes the associated hash value. Confirming the equality of two potentially large data sets can be achieved by simply comparing the root hashes of the trees.

During block verification, the verifiers take the resulting world-state of the previous block, execute all transactions (including contract executions) and compare the hash of the updated world-state to the hash included in the proposed block header. This is the essential operation common to all consensus mechanisms.

<u>**Off-chain Operations:**</u> Blockchain applications in the recent age have been managing data either as on-chain or off-chain as storage mechanisms. **On-chain operations focus more on transparency and audibility to transactions and the information stored onchain.** Blockchain-

based transactions can be off-chain bringing great benefits in increasing security but also release the mining time speed limit. **This is because off-chain transaction does not require nodes to conform/consensus, and meanwhile, off-chain makes the system avoid public and open network. Off-chain operations for blockchain can be faster, cheaper and provide more privacy.** The off-chain solutions should reduce data storage on-chain, computational and financial cost.

- *Offchain payment.* Transactions can be off-chain transferred, like lightning network [52] – a second layer to Bitcoin for micropayment between two parties off-chain, and Raiden (raiden.network) for Ethereum. Off-chain transactions have been studied with smart contract in Commit-Chains [53], Sprites, and BRICK [54] in which smart contract is used to verify status and enforce correct behaviours. But the first two do not consider privacy – value/transaction privacy and anonymity. [55] introduces an anonymous off/on-chain payment on Bitcoin platform.
- *Off-chain storage.* Off-chain storage mode [56] should be also considered while using smart contract to deal with large amount of storage data. On-chain storage is extremely expensive here. Smart contract may define a hash value (of the file, stored on-chain) point to off-chain reference, like a pointer, for further data retrieval from off-chain storage system. Data integrity may be checked via the hash value on-chain. Interplanetary File System (IPFS) [57], Swarm [58] and Sia (https://sia.tech/) are currently employing the above philosophy in off-chain content storage. Some applications may direct combine distributed databases with blockchain for efficient storage, e.g., EthDrive [59] and BigchainDB [60]. DataVaults will need to examine how to perform efficient offchain data search and meanwhile, preserve the security of search contents.
- *Offchain smart contract verification.* The verification of off-chain computation from smart contract enables one to execute heavy computation offline and provide a non-interactive verification for the computation. Classical examples can be seen in zero-knowledge Succinct Non-interactive ARgument of Knowledge (zkSNARKs) [61, 62], Bulletproof [63] (suitable for a range proof), and Zero-Knowledge Scalable Transparent ARguments of Knowledge (zkSTARKs) [64]. These three mechanisms have pros and cos: for prover and verification time, zkSNARKs and zkSTARKs are much faster than Bulletproofs, but Bulletproof and zkSNARKs have smaller proof size. zkSNARKs require trusted setup to achieve strong crypto assumption, while zkSTARKs and Bulletproof do not need trusted setup relying on collision resistance and discrete log hard problems. These verifications may be considered using on DataVaults smart contract in the project, depending on the needs from use case partners.

3.4.1 Consensus

The concrete implementation of the consensus system is responsible for agreeing on new blocks by including transactions in an unbiased and clearly specified way. In most public Blockchains this is achieved with a Proof of Work (PoW) algorithm, where participants need to find a random nonce value to produce a hash value with a certain format. This is a computational intensive process, but the result is easy to verify by the other nodes.

For enterprise Blockchains, this system would be too inefficient and limit the throughput artificially. With the permissioned access, there can also be put a higher trust into the correct behaviour of the nodes. Usually a *leader* is elected by voting or assigned in a round-robin sequence. It proposes a candidate block which is distributed to the verifier nodes for voting.

If a sufficient number of nodes accept the block, it is appended to the chain head. Usually there are also measures in place to remove the authority from faulty or misbehaving nodes. **For the DataVaults platform, every consortium member could contribute as a verifier. With seventeen independent partners sharing the liability of providing a secure platform, the correctness is assured on a high security level.**

To enable the flexible, secure and light-weight-but-still-scalable design, DataVaults may consider combining the use of PBFT and Hyperledger family consensus algorithms (note the most current one is the RAFT [41]).

Quorum (Section 4.3) offers a selection of three different consensus mechanisms:

- Raft [6] features low transaction latency (50 ms minimum block period) with on-demand forging of new blocks. Provides efficient consensus when nodes are trusted, with the downside of only providing crash fault tolerance (CFT).
- The Istanbul Byzantine Fault Tolerant (IBFT) consensus implementation is a variant of Practical Byzantine Fault Tolerance (PBFT) [7]. It is robust against malicious nodes at the cost of exponentially increasing network overhead with the number of participants.

Clique Proof of Authority (PoA) [8] features a simple specification for managing a list of authorized nodes. They can arbitrarily create blocks with a basic arbitration scheme preventing collisions (e.g. round-robin). This mechanism is mainly used for implementation and testing of applications because of the reduced complexity.

4 DATAVAULTS ENHANCED DATA PRIVACY MECHANISMS

4.1 DATAVAULTS VALUE PROPOSITION & CONCEPTUAL ARCHITECTURE

As described in the previous chapter, the creation of a digital data marketplace, based on the use of Blockchain Distributed Ledgers is considered the main value proposition of DataVaults. However, besides only security, **privacy** is also considered one of the core requirements that must be managed efficiently together with **scalability, smart contract verification, data storage, consensus mechanisms, etc.** Taking into consideration that most users should be disenfranchised from this process since it cannot be expected that they will have a clear understanding of the various data security and user privacy implications, it is imperative to build new **on- and off-chain data management models and services** of privacy and data protection and the technologies that encode them.

In this direction, Datavaults enables enhanced **data privacy** and ownership safeguarding (**privacy by design**) and **data provenance and sovereignty** checking mechanisms. The platform uses Blockchain-based distributed ledgers for offering enhanced data and transaction security. Blockchain is one of the most disruptive technologies related to data security today, but beyond the inherently sensitive nature of various personal and commercial data are the persistent challenges of interoperability, data matching, and data information processing, sharing and exchange. To this end, DataVaults protects data and resources against leak or improper modifications, while at the same time ensures data availability to legitimate users. Internal storage and ledger infrastructures, handling personal and/or corporate data, can track its provenance and are regularly audited to comply with specified **security and privacy policies and regulations**. This way users are in control of their own privacy and that of their devices, applications and services. For the former, users will be able to participate in the specification of privacy-related policies, afterwards translated in the appropriate smart contracts, following the principle of user privacy empowerment. Depending on the selected privacy level, **privacy enhancement is achieved through the use of trusted computing technologies** (i.e., TPMs) as a central building block towards the provision of privacy-preserving signature schemes (Direct Anonymous Attestation (DAA). **By assuring auditable, security and privacy policy compliant actions, DataVaults also guarantees that application ecosystems where such policies have been technically enforced are highlighted.**

As put forth in Section 3.1, DataVaults will leverage two general types of ledger infrastructure, namely a private ledger which is responsible for the **creation and validation of contracts between the DataVaults Platform and the Individual**, based on the details of the data sharing transactions, and a public ledger for **capturing and recording the fine-grained details of extracted metadata towards efficient data search** (Figure 14).

Reflecting on DataVault's work and data flow and how provided data security, privacy, sharing and management services can be engrained into the policy-compliant DataVaults structure, the envisaged conceptual architecture (Figure 14) captures the following set of provided **on- and off-chain control functionalities** and services based on the use of hardware trust anchors for privacy-preserving data trading services:

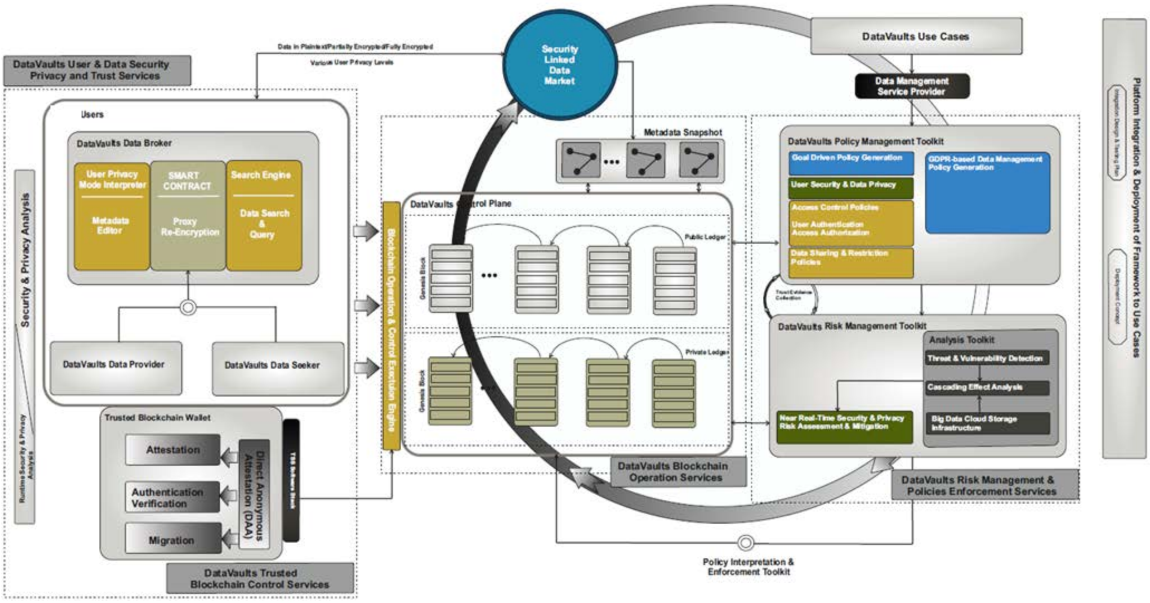

Data in Plaintext/Partially Encrypted/Fully Encrypted
Various User Privacy Levels
DataVaults User & Data Security Privacy and Trust Services
Users
DataVaults Data Broker
User Privacy Mode Interpreter
Metadata Editor
SMART CONTRACT
Proxy Re-Encryption
Search Engine
Data Search & Query
DataVaults Data Provider
DataVaults Data Seeker
Security & Privacy Analysis
Runtime Security & Privacy Analysis
Trusted Blockchain Wallet
Attestation
Authentication Verification
Migration
Direct Anonymous Attestation (DAA)
TSS Software Stack
DataVaults Trusted Blockchain Control Services
Blockchain Operation & Control Execution Engine
Security Linked Data Market
Metadata Snapshot
DataVaults Control Plane
Genesis Block
Public Ledger
Private Ledger
DataVaults Blockchain Operation Services
DataVaults Use Cases
Data Management Service Provider
DataVaults Policy Management Toolkit
Goal Driven Policy Generation
GDPR-based Data Management Policy Generation
User Security & Data Privacy
Access Control Policies
User Authentication Access Authorization
Data Sharing & Restriction Policies
Trust Evidence Collection
DataVaults Risk Management Toolkit
Analysis Toolkit
Threat & Vulnerability Detection
Cascading Effect Analysis
Big Data Cloud Storage Infrastructure
Near Real-Time Security & Privacy Risk Assessment & Mitigation
DataVaults Risk Management & Policies Enforcement Services
Policy Interpretation & Enforcement Toolkit
Platform Integration & Deployment of Framework to Use Cases
Integration Design & Testing Plan
Deployment Concept

Figure 14: DataVaults Blockchain-based Conceptual Architecture

DataVaults Trusted Blockchain Control Services: Trusted Platform Modules (TPMs) are a central building block of DataVaults privacy-preserving mechanisms and form the basis for enhanced security, privacy and reliability guarantees for ledger management and maintenance. The smart integration of the TPM technology will allow DataVaults to develop new Blockchain verification methods and significantly advance the state-of-the-art of Blockchain operation services: (i) **secure storage**: a user can store any secrets (keys, passwords or other sensitive data) associated with a TPM, and, when authorized by the user, the TPM allows access to the user's secrets, and (ii) **secure execution**: it provides a trusted execution environment that allows the isolated, secure execution of code mainly for protecting the execution of security-relevant code.

Trusted Blockchain Wallet: In the DataVaults framework, TPMs are also the basis for trusted Blockchain wallets. They will be used to: (i) provide strong user authentication and to securely store the user credentials based on the TPM's secure key storage, (ii) control and authorize access to *private* or *public* ledger channels based on the user authentication process (e.g., to authorize access to or operations on different ledgers), and (iii) securely and efficiently verify Blockchain updates. In this way, DataVaults will significantly advance the state-of-the-art of Blockchain verification methods: Unlike current mechanisms that often rely on computationally costly and wasteful proofs of work or biased proofs of stake**, DataVaults will use TPMs as central building block to build a very resource-efficient and trustful two-staged Blockchain verification mechanism, which will be even suitable for resource-constrained devices** (such as smart devices - equipped with a TPM). From a TPM perspective, the continuous verification procedure of Blockchain edits can be outlined as follows, where we will assume that all participating entities hold the current Blockchain state hash inside their TPMs: In Stage 1, the data broker will perform a pending Blockchain update, and will then determine the updated Blockchain state hash based on the ledger updates and the current state hash. Then, in Stage 2, the chosen verifiers (and any other DataVaults users) are able to verify the update. This involves checking the validity of the updated blockchain state based on the block update and the current blockchain state hash. On success, the users will then replace the current state hash inside their TPMs with the updated one.

Trusted Blockchain Attestation: In order to guarantee that only **trusted and uncompromised devices can participate in DataVaults**, all involved devices will use the TPM secure boot mechanism and their trust level will be continuously attested and assessed. To this end, all signatures on DataVaults data (e.g., transactions, smart contracts) will include the respective platform's integrity state (which is the hash value held by the device's PCRs at the end of the secure boot process), which will allow any other party to check whether the data stems or was acknowledged by a trusted DataVaults user. **Depending on the selected privacy level, a conventional or a privacy-preserving signature scheme may be employed**. In the former case, a plain digital signature scheme supported by the TPM (e.g. ECDSA) will be selected, whereas in the latter case the TPM-provided DAA scheme can be used as strong privacy-preserving signature scheme. DAA (Section 4.2.3) can provide anonymous authentication, attestation and date integrity services. Several DAA schemes and their applications are specified in ISO/IEC 20008 and ISO/IEC 20009, respectively.

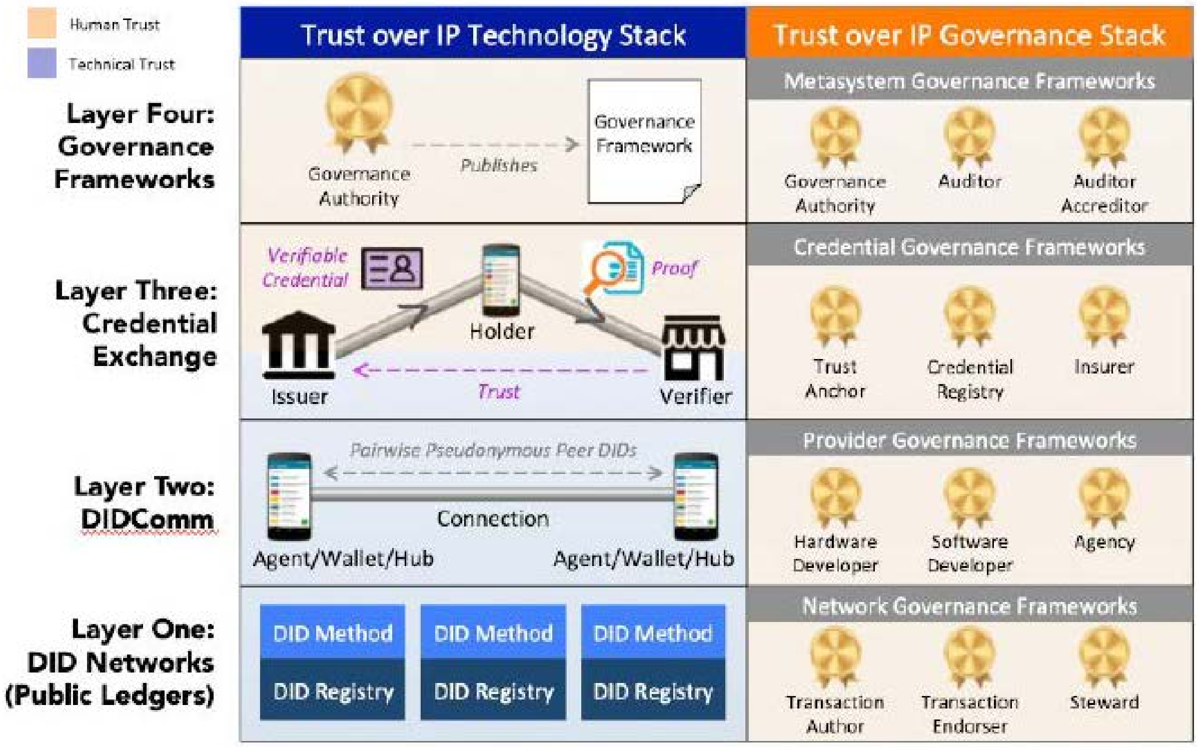


Figure 15: Trust and Privacy Layers in DataVaults

Trusted Authentication: To secure communication and prevent impersonation and man-in-the-middle attacks, peer authentication is of extreme significance. DataVaults will offer **multi-tier secure authentication** based on the aforementioned hardware root-of-trust (Figure 15): (i) trusted identity authentication between peers, (ii) trusted **membership authentication** for read and write on ledger, (iii) trusted **access authentication** for cloud-cased storage system, and (iv) trusted **actioner authentication for data search and sharing**. DataVaults guarantees that a user or a party claims what it is that is exactly what it is, which means that trust can be delivered inside the physical level – providing trustworthiness for the device managed by the user.

4.2 SECURITY AND TRUST BUNDLES FOR USER PRIVACY AND CONVEYANCE OF DATA

In what follows, we will present and assess the cryptographic primitives and protocols leveraged by DataVaults for achieving enhanced user privacy protection - needed to secure different types of information, while still allowing advanced knowledge discovery through the provision of **enhanced data search services** (i.e., Searchable Encryption), and **advanced security and privacy-preserving primitives** (i.e., data anonymization and pseudonymization techniques) for authentication, authorization, attestation and verification through the use of trusted computing technologies. Such an analysis will serve as the basis (and provide valuable insights) on the definition of the overall DataVaults conceptual architecture and identification of all internal interfaces to be documented in D5.2 [16].

4.2.1 Attributed-based Encryption for User Privacy and Conveyance of Data

Attributed-based Encryption (ABE) is an encryption concept introduced in 2004 by Sahai and Waters [11], and the main idea is to allow a user to encrypt data that it can only be decrypted by users with certain attributes. **Attribute is a Characteristic of an object or entity, and in ABE the attributes are the characteristics that can be used to define who should be able to decrypt a ciphertext and who should not.** ABE is a generalization of **Identity Based Encryption (IBE)** where the encryption and decryption is performed with a single attribute; **the data provider identity.** Two main types of attribute-based encryption schemes exist [12], the **key-policy attribute-based encryption (KP-ABE)** and the **ciphertext-policy attribute-based encryption (CP-ABE)**.

CP-ABE allows the encryptor to determine which users are able to decrypt certain ciphertexts by setting the access policy when generating said ciphertexts. This is opposed to KP-ABE where the key issuer determines which policy is used to generate the key and so there is an additional need to trust the key issuer.

ABE is an ideal solution for addressing the problem of data access and revocation, as revoking can be performed based on attributes (e.g., by a time attribute). **ABE based schemes allow user to encrypt a file based on a certain policy and provide a unique key is that is generated based on a list of attributes, for each user that has access resources.** Then a user is able to decrypt a file that is associated with a certain policy only if the attributes of her key satisfy the underlying policy [12].

Most of the ABE schemes consists of four basic steps:

- **Setup:** this step is the initialization of the algorithm to produce to data structures; a public key (PK) used to encrypt and generate decryption keys, and the Master Key (MK) used to generate Decryption Keys (DK).
- **Encryption:** this step produces the corresponding cyphertext of a document, taking as input the document, PK and the policy that must be fulfilled in order to decrypt the cyphertext. Ideally it can be seen as the policy being incrusted into the cyphertext.

- **Key Generation:** in this step, a requester must present a set of attributes which are used to generate a DK.
- **Decryption:** In this step the cyphertext is computed with DK to be decrypted. If the attributes used to generate DK fulfils the policy incrusted in the cyphertext the decryption is successful.

ABE schemes enable to cypher a document one time with a policy and create as many DK as available attribute combinations there are. Later the cyphered document can be sent to every user and only those who owns a DK with the appropriate attributes will be able to decrypt it.

It must be noted that when talking about attributes, and from the point of view of encryption algorithm, it means only **values of attributes.** Therefore, it is necessary to agree in a common attribute model to be applied to both, policies (used in the encryption by data owners) and user attributes (used to generate DK for decryptors).

In the case of DataVaults, **this model should be enough to enable the user to manage to access or use her/his data, but no so specific as to filter by personal details.** This is, the data owner should be able to manage details like period of accessibility, aim of the access, type of organization that can access data (public, private, NGO, health, etcetera), but not details as name of the person how access the data, because the data owner does not know that details in advance.

The management of the attribute model introduces the idea of centric solution in order to make it public available for all actors in DataVaults, even more if we consider this model as live entity. It can change due to evolution of the model, and also if we consider that a change in the model is one of the easiest ways to revoke DK.

An example of this is to change how to express dates, numbers instead of names to identify months. Keys generated with number of the month as attribute will not decrypt cyphertexts encrypted with names and vice versa. Also, some ABE encryption models enable advance users to extract attributes from DK. It constitutes a weakness which can be prevented if the model is changed periodically.

The strength of ABE encryption schemes resides in the fact that a document needs to be encrypted one only time and can be deployed to any data seeker. Only those data seekers with the appropriate DK will get access to the document. Therefore, the main task resides in the step of Key Generation. The system must ensure that any user requesting a DK will obtain this key based on the attributes she/he legitimately owns. To ensure this point there are two main alternatives:

- The user requesting a DK is known and therefore attributes were set in advance, for instance integrating identification and authorization mechanisms.
- The user is not known but presents a list of attributes endorsed by a trusted authority.

Regarding keys management, the entity in charge of setup and key generation steps is usually named Master Authority (MA). Depending on the business case there are two main variants, central solutions and decentralized solutions.

In central solutions there is one single Master Authority entity. It performs creation of MK and PK for each user, and generation of DKs for each decryptor, and besides, it is used to implement hosting of MKs and key deployment (PKs, DKs). The main advantage of centric solutions is that the MA provide trust over the whole encryption system. But at the same time is the main weakness point as it can create keys to access to any cyphertext by itself and violate privacy of users.

In contrast, decentralized solutions enable any user to become central authority, and create her/his own MK and PK. **From the point of view of privacy this is the most suitable variant but can result in overload of the user as it must also take tasks of creation and deployment of DKs.** Moreover, it is the user who must host MK and provide key recovery functionality. In this case we can consider that for single users this is the weakest variant. Besides these considerations, decentralized solutions need a common attribute model to enable all players to use same language. Therefore, the centricity flavor persists.

Considering the usage of ABE in the scope of DataVaults, the first part is related to the encryption of the files. Although many details are still to be defined, the figure below provides an implementation of the ABE flow. ABE flow is initiated upon a proper initialization phase (**Step-1**) according to which a public file and a master-key-generator are produced.

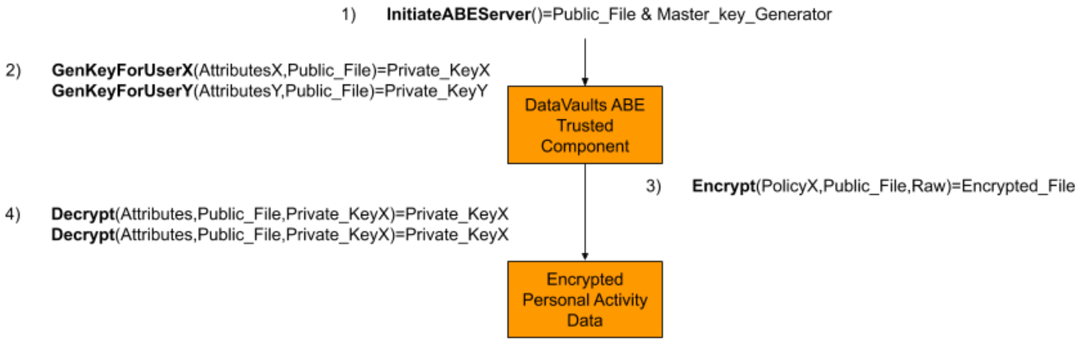

1) **InitiateABEServer**()=Public_File & Master_key_Generator
2) **GenKeyForUserX**(AttributesX,Public_File)=Private_KeyX
GenKeyForUserY(AttributesY,Public_File)=Private_KeyY
DataVaults ABE Trusted Component
3) **Encrypt**(PolicyX,Public_File,Raw)=Encrypted_File
4) **Decrypt**(Attributes,Public_File,Private_KeyX)=Private_KeyX
Decrypt(Attributes,Public_File,Private_KeyX)=Private_KeyX
Encrypted Personal Activity Data

Figure 16: Basic Attribute-Based Encryption Workflow

Upon initiation, many users can ask the ABE Trusted Component (that holds in a protected zone the master key) to issue a private key based on a set of attributes that are verifiable (e.g. firstname: x1, organization:org1) (**Step 2**).

Each party can encrypt a document using the ABE-Server’s public key and a set of attributes; the attributes should match in order to decrypt a file. A policy can be {firstname:y1 or organization:org1}. The policy is encrypted along with the rest of the raw file. Two independent users can attempt to decrypt the file based on their key without having proper knowledge whether their keys can perform decryption or not.

Furthermore, the usage of ABE in the scope of DataVaults and the DataVaults lifecycle, we consider the **close integration of the Smart Contracts and the ABAC based data access control with ABE**; this is mainly based on the fact that ABAC also provides data access based on attributes and policies. **ABE is not an authorization mechanism but a cryptographic primitive that allows multiple users to encrypt and decrypt files based on their attributes and encryption policies**, however it has some inherent authorization properties since it allows the definition of (encrypted) policies that mandate whether or not a user-key can decrypt a cypher or not.

For this reason, we consider a **combined use of ABAC and ABE policies**, where an ABAC policy based is applied at first step and controlling the access to data, and then if an ABAC permit is granted, the second step is to apply an ABE policy in order to decrypt **the data or the resource symmetric decryption key**. For the scope of DataVaults we will consider the usage of ABE to the encryption and decryption of the data and the keys. Additionally, we **consider the usage of TPM for the key hierarchies and the management of the aforementioned secret keys.**

In addition to this, an important part of the ABE is the **verifiability of attributes**; PM is also considered for the verifiability of the attributes, in the case that the attributes are provided by the client. An option to retrieve the attributes for both ABAC and ABE is also the usage of a centralized server that can provide electronically signed attributes that can be used by the ABAC authorization engine.

The envisioned flow for integrating the ABE with the data access control of DataVaults is depicted in figure below (Figure 17).

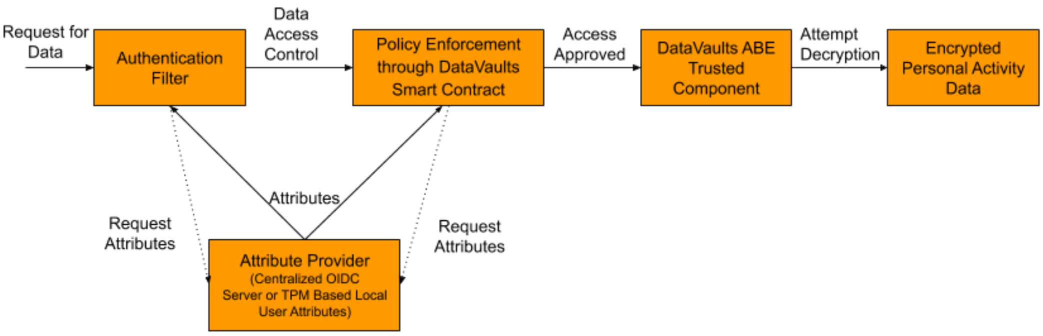

Request for Data
Authentication Filter
Data Access Control
Policy Enforcement through DataVaults Smart Contract
Access Approved
DataVaults ABE Trusted Component
Attempt Decryption
Encrypted Personal Activity Data
Attributes
Request Attributes
Request Attributes
Attribute Provider
(Centralized OIDC Server or TPM Based Local User Attributes)

Figure 17: Steps for the Authorization process

Initially, a data seeker is performing a request in order to access a specific encrypted data. Before the evaluation of whether or not the data seeker is allowed to access the data the request per se will be intercepted by an authentication filter based on the Smart Contracts. We consider the usage of two different ways for providing attributes. The first is that user is providing local user attributes through a TPM enabled trusted device, or through a centralized authentication proxy that is providing both the identity and a set of electronically signed attributes that are verifiably associated with the user per se.

The initial request is intercepted by the Policy Enforcement element that identifies which policies are relevant for this request, which attributes are relevant for these policies and finally evaluates the policy expressions based on the values of these policies.

During the evaluation of the expression, some rules may advise towards allowing and some others towards denying the requests. Upon allowance, the flow continues with an attempt for ABE based decryption using the DataVaults ABE Trusted Component. We refer to the term “attempt” because the attributes that have been used for the encryption policy of the resource may indicate that this user although s/he can access the resource s/he cannot decipher it.

4.2.2 User PERSONAS

As discussed in the WP1 deliverables and in the overall DataVaults concept, the platform will offer the option to individual users to share their data assets both in **a non-anymised and in an anonymised manner, based on their preferences.** When it comes to the second case (anonymously sharing data), two different methods are envisaged; building an anonymous digital twin of a user, thus, pseudonymising the user but essentially keeping most of its data intact to allow more precise data analysis operations by data seekers, and that of anonymising and obfuscating user's data and merging them with data from users with similar characteristics, constructing at the end a fictional "User Persona".

User Personas therefore have the ability to further mask the real data of individuals, maximising their privacy as the data seeker at the end is receiving at his hands numbers which represent the group of the individuals that belong to this persona, and not the actual set of numbers of all individuals under this persona. At the same moment, as data is masked, it is obvious that it loses some of its value, and the individual who shares data through this mechanism should anticipate less rewards.

The following figure provides a high-level description of how a User Persona under DataVaults is constructed.

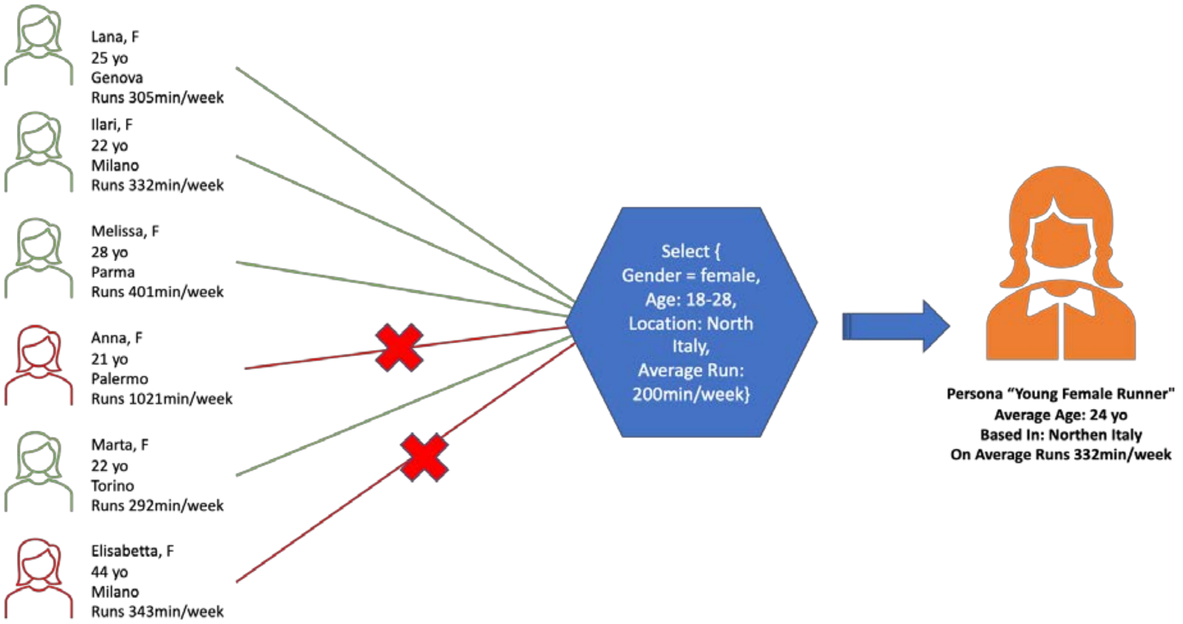


Figure 18: High Level Persona Concept

As Figure 15 depicts, a Persona named "Young Female Runner" is constructed by building a query that runs within a pool of shared individuals' data who comply with the following criteria

- Are females;
- Are aged between 18 and 28;
- Are based in North Italy;
- Run on average 200 mins per week;

As can also be seen in the figure, the individuals (which have provided their consent to share their data under personas) which comply with those query parameters are Lana, Ilari, Melissa and Marta, while Anna and Elisabetta are not part of the Persona (as the former resides in South Italy and the latter is more than 28 years old). At the end, the result would be the Persona "young Female Runner" that would include also other data which are shared by those individuals (for example average weight, heartrate, online activity, marital status, etc.) which match the query parameters and would therefore constitute a representation of a group of females that are based in North Italy and run at least 200 mins per week.

In order to be able to construct a Persona, the following criteria must be present:

- There should be **users with similar characteristics**, which are those defined by the initial query that is used to construct the User Persona.
- There should be an **adequate number of individuals with similar characteristics** (e.g. the query results mentioned above), in order to form a persona that makes it difficult to backtrack individuals. The rationale behind this requirement is that in case a persona with very few individuals is constructed (for example with 3 individuals), then it is easier for a malicious user to identify those 3 individuals in case he combines the persona knowledge with other knowledge that may come from the platform or from other sources. **This number is set by the platform and should be more than 20 in the beginning.**

Also, it needs to be mentioned that User Personas should **not be static representations of a group of individuals**, as the concept behind them is to evolve and behave as individuals evolve. For this reason, we consider **User Personas as dynamic representations of individuals.** For this reason, Personas are updated in regular intervals, to acquire the updated data from individuals (for example to update the average heartrate of "*Young Female Runner*" per day based on the data recorded by each individual, while at that point also the Persona checks whether an individual shall be included in the Persona as part of a "Join Request", or stop from being a member of it, due to a "Leave Request" or an "Auto Expel" decision.

In this context, the following three mechanisms for joining and leaving a persona take place.

- **Join Request:** An individual can automatically join a Persona in case he/she has selected to share similar data with that of the persona, and whose data are within the boundaries of the query which instantiated the persona. For example, Lara who is a female of 25 years age and runs every week for an average of 305 minutes can automatically join the Persona "*Young Female Runner.* However, she cannot join the Persona "*Middle Aged Male Runners*", as this would be a Persona that concerns males of at least 40 years of age.
- **Leave Request:** An individual can at any time request to leave a Persona by selecting not to share relevant data. In this case, the **data of the individual are not considered as part of updated version of the Persona.**
- **Auto Expel:** An individual whose data deviates from the values set for defining the Persona are automatically expelled from that Persona. For example, if Lara in the previous example abandons her running routine and starts running for less than 200 minutes per week, she will not be included in the Persona when the latter automatically updates.

4.2.3 Direct Anonymous Attestation (DAA)

For privacy, DataVaults will also offer another variant (on top of the option for creating User PERSONAS) by leveraging advanced crypto primitives, namely **Direct Anonymous Attestation (DAA)** [48, 49] based on group signatures. Privacy requirements that are captured by DAA are the ones already documented in the ETSI TS 102 941 standard [50]: **anonymity** *(ability of a user to use a Datavaults resource without disclosing its identity)*, **pseudonymity** *(ability of a user to use a DataVaults resource without disclosing its identity while being accountable for that action)*, **unlinkability** *(ability of a user to make multiple uses of DataVaults resources without others being able to link them together)*, **and unobservability** *(ability of a user to use a DataVaults resource without others being able to observe that the resource is being used)*.

In this context, **the actual identity of the data provider is not required for ensuring the trustworthiness of a transmitted message.** It rather suffices to verify the **origin correctness**; a message has been sent by a valid "data provider". Indeed, since exchanged messages might contain sensitive data, what is required is that certificates should not contain any identifying information that could trace them back to a particular device or platform. In this context, **DataVaults leverages anonymous credentials through the use of Direct Anonymous Attestation (DAA) addressing all the aforementioned limitations, i.e., privacy, security, and accountability.**

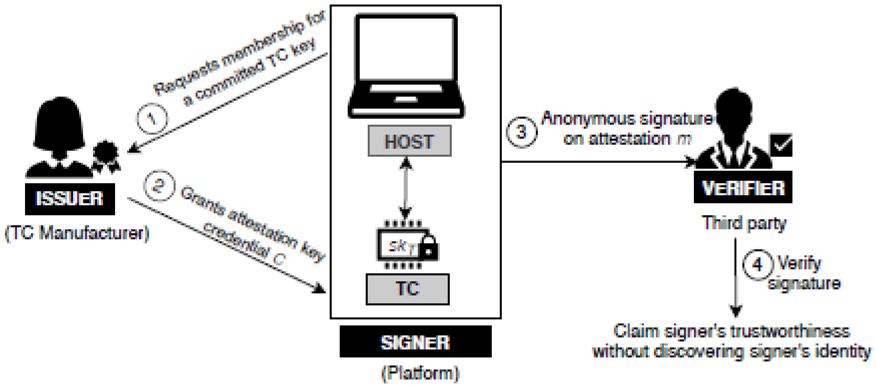

Requests membership for a committed TC key
1
ISSUER
(TC Manufacturer)
2
Grants attestation key credential C
HOST
sk_T
TC
SIGNER
(Platform)
3
Anonymous signature on attestation m
VERIFIER
Third party
4
Verify signature
Claim signer's trustworthiness without discovering signer's identity

Figure 19: Entities involved in the DataVaults DAA Protocol

DAA is a platform authentication mechanism that enables the **provision of privacy-preserving and accountable authentication services.** DAA is based on group signatures that give strong anonymity guarantees. The key security and privacy properties of DAA are:

- *User-controlled anonymity*: Identity of user cannot be revealed from the signature;
- *User-controlled linkability*: User controls whether signatures can be linked;
- *Non-frameability*: Adversaries cannot produce signatures originating from a valid trusted component;
- *Correctness*: Valid signatures are verifiable, and linkable, where needed.

A DAA scheme considers a set of Issuers, hosts, Trusted Components (TCs - TPMs in the context of DataVaults), and verifiers (Figure 19); the host and TC together form a trusted Data Provider node. In the context of DataVaults, the TC functionalities will be offered by the underlying Blockchain Starter Kit. The Issuer is a trusted third-party (DataVaults Cloud Platform) responsible for attesting and authorizing platforms to join the system (through the execution of zero-touch configuration integrity verification process). A verifier is any other system entity or the Datavaults Cloud Platform itself that can verify a platforms' credentials in a privacy-preserving manner using DAA algorithms; without the need of knowing the platform's identity. The Elliptic-curve cryptography (ECC) based DAA is comprised of five algorithms SETUP, JOIN, SIGN, VERIFY and LINK (Figure 17).

- **SETUP:** The system parameters must be chosen and the Issuer needs to generate its keys. The system parameters and the Issuer's public keys are then published and available to the cluster and to anyone who needs to verify the validity of a signature.
- **JOIN:** A Host using a TC joins the group and obtains an Attestation Key Credential (AKC) for an ECC-DAA key created by the TC. The key can then be used to anonymously sign a message, or attest to data from this TC.
- **SIGN:** Using the ECC-DAA key, for a range of signing operations.
- **VERIFY:** Verifying a signature and returning true (valid) or false (invalid).
- **LINK:** Checking two signatures to see if they are linked and returning true (linked) or false (un-linked).

A DAA scheme enables a data provider to prove the possession of the issued credential C (access token) to the Datavaults Cloud Platform by providing a signature, which allows the platform to authenticate the data provider without revealing the credential C and provider's identity. In a nutshell, DAA is essentially a two-step process where, firstly, the **registration of a data provider executes** and during this phase the user chooses a secret key (SETUP). This secret key is stored in secure storage so that the host cannot have access to it – it can be accessed through appropriate interfaces offered by the Blockchain Starter Kit. **Next the user talks to the platform so that it can provide the necessary guarantees of its validity** (JOIN). The platform then places a signature on the public key, producing the Attestation Identity Credential (AIC) <cre>. The second step is to use this <cre> for **anonymous attestations** on the platform (SIGN), using Zero-Knowledge Proofs [51]. **These proofs convince a verifier that a message is signed by some key that was certified by the issuer, without knowledge of the TC's DAA key or <cre> (VERIFY).**

However, we have to highlight that in Blockchain-based environments, DAA cannot be directly used to preserve the privacy of the users without performing some necessary updates regarding the management of the DAA Key. **Quorum uses a key-pair for signing the transactions, permissions and receiving of funds.** It is publicly broadcasted with the transaction to allow other nodes to verify the authenticity. Individual transaction can, thus, be linked to the same account which breaches the fundamental properties of DAA. Another issue arises with the used crypto primitives: Most hardware TPMs use the *NIST P-256* (secp256r1) parameters for Elliptic Curve Cryptography (ECC), whereas Quorum requires the *Koblitz* (secp256k1) parameters for signing of the transactions.

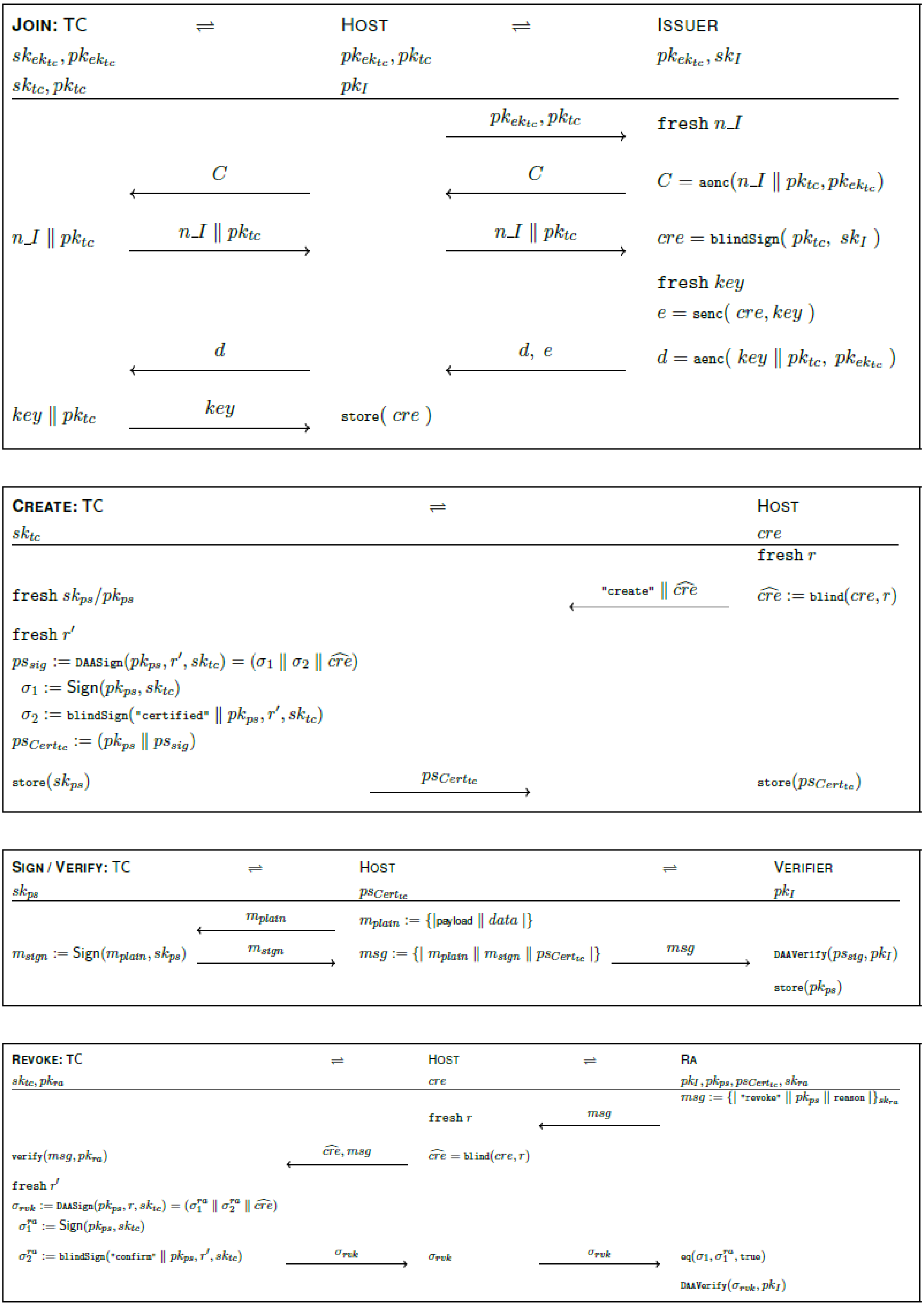

JOIN: TC ⇌ HOST ⇌ ISSUER
$sk_{ek_{tc}}, pk_{ek_{tc}}$
sk_{tc}, pk_{tc}
$pk_{ek_{tc}}, pk_{tc}$
pk_I
$pk_{ek_{tc}}, sk_I$
$pk_{ek_{tc}}, pk_{tc}$
fresh n_I
C
C
$C = \texttt{aenc}(n\_I \parallel pk_{tc}, pk_{ek_{tc}})$
$n\_I \parallel pk_{tc}$
$n\_I \parallel pk_{tc}$
$n\_I \parallel pk_{tc}$
$cre = \texttt{blindSign}(\ pk_{tc},\ sk_I\ )$
fresh key
$e = \texttt{senc}(\ cre, key\ )$
d
$d,\ e$
$d = \texttt{aenc}(\ key \parallel pk_{tc},\ pk_{ek_{tc}}\ )$
$key \parallel pk_{tc}$
key
$\texttt{store}(\ cre\ )$
CREATE: TC ⇌ HOST
sk_{tc}
cre
fresh r
fresh sk_{ps}/pk_{ps}
$\texttt{"create"} \parallel \widehat{cre}$
$\widehat{cre} := \texttt{blind}(cre, r)$
fresh r'
$ps_{sig} := \texttt{DAASign}(pk_{ps}, r', sk_{tc}) = (\sigma_1 \parallel \sigma_2 \parallel \widehat{cre})$
$\sigma_1 := \mathsf{Sign}(pk_{ps}, sk_{tc})$
$\sigma_2 := \texttt{blindSign}(\texttt{"certified"} \parallel pk_{ps}, r', sk_{tc})$
$ps_{Cert_{tc}} := (pk_{ps} \parallel ps_{sig})$
$\texttt{store}(sk_{ps})$
$ps_{Cert_{tc}}$
$\texttt{store}(ps_{Cert_{tc}})$
SIGN / VERIFY: TC ⇌ HOST ⇌ VERIFIER
sk_{ps}
$ps_{Cert_{tc}}$
pk_I
m_{plain}
$m_{plain} := \{\mid \mathsf{payload} \parallel data \mid\}$
$m_{sign} := \mathsf{Sign}(m_{plain}, sk_{ps})$
m_{sign}
$msg := \{\mid m_{plain} \parallel m_{sign} \parallel ps_{Cert_{tc}} \mid\}$
msg
$\texttt{DAAVerify}(ps_{sig}, pk_I)$
$\texttt{store}(pk_{ps})$
REVOKE: TC ⇌ HOST ⇌ RA
sk_{tc}, pk_{ra}
cre
$pk_I, pk_{ps}, ps_{Cert_{tc}}, sk_{ra}$
$msg := \{\mid \texttt{"revoke"} \parallel pk_{ps} \parallel \texttt{reason} \mid\}_{sk_{ra}}$
fresh r
msg
$\texttt{verify}(msg, pk_{ra})$
$\widehat{cre}, msg$
$\widehat{cre} = \texttt{blind}(cre, r)$
fresh r'
$\sigma_{rvk} := \texttt{DAASign}(pk_{ps}, r, sk_{tc}) = (\sigma_1^{ra} \parallel \sigma_2^{ra} \parallel \widehat{cre})$
$\sigma_1^{ra} := \mathsf{Sign}(pk_{ps}, sk_{tc})$
$\sigma_2^{ra} := \texttt{blindSign}(\texttt{"confirm"} \parallel pk_{ps}, r', sk_{tc})$
σ_{rvk}
σ_{rvk}
σ_{rvk}
$\texttt{eq}(\sigma_1, \sigma_1^{ra}, \texttt{true})$
$\texttt{DAAVerify}(\sigma_{rvk}, pk_I)$

Figure 20: High-level Overview of the DataVaults DAA Protocol Interfaces

To solve this issue, a two-step approach will be used: A secure channel is established with the help of DAA, meaning the users authenticate themselves and exchange session keys. This provides a proof to the platform that the user controls a valid TPM, issued with access permissions. For previously unused DAA-identities, also a fresh Blockchain key-pair needs to be generated and send to the platform for authentication. The user can then use this new, privacy preserving identity to use all Blockchain related services.

4.3 INTEGRATION WITH DATAVAULTS DISTRIBUTED LEDGER TECHNOLOGY

As aforementioned, the overall goal of DataVaults is the provision of a secure, trusted and auditable data sharing environment based on the use of **distributed ledger technologies and signed smart contracts** to capture data sharing (while complying with the prevailing GDPR legislation), collection, and compensation and trading preferences among the DataVaults parties. In this context, the Distributed Ledgers Technology to be used is an integral part for connecting and achieving all the previously described requirements and envisioned functionalities. **While some properties are already covered by the basic features of most Blockchain implementations, the selection of applicable products is further restricted by the advanced requirements for privacy and security.**

In what follows, we summarize the main considerations that were discussed within the consortium towards choosing the more appropriate DLT technology to be adopted –based on the above but also the high-level comparison of various Blockchain environments and tools that was put forth in D2.1 [13]. ***Quorum* was selected as the target technology, because it already supports most of the envisioned functional requirements by default.** For features that are not yet fully available, a preliminary investigation was performed to estimate the difficulty of their implementation and integration in the overall platform.

All fundamental features of the Blockchain technologies presented in D2.1 are also supported by Quorum. **The core of the system is provided by a fork of the popular *Ethereum* Blockchain. Additional modules were added to provide a solution for enterprise use cases.** Some of the advanced functionalities are identified below as main differentiators to other products – these also enable some of the previously described innovative features of DataVaults.

- **Private, permissioned network:** Network access is configured with a **Role-Based Access Control (RBAC) Mechanism.** The on-chain configured permissions are used to restrict read, transaction and contract deployment operations to authorized users. It is implemented with a smart contract design which provides functionalities for managing organisations and voting on new nodes.
- **Selectable consensus:** Most public Blockchain systems use an inefficient Proof-of-Work (PoW) consensus to guarantee integrity. With a permissioned system, more sophisticated algorithms can be used to reach consensus. Based on the estimated thread model, one of many algorithms can be selected. The two most popular options for Quorum are **RAFT or IBFT** (Section 3.2.6), whereas the former only provides Crash Fault Tolerance (CFT) and the later also covers Byzantine Fault Tolerance (BFT).
- **Smart contracts:** Based on the close relation to Ethereum, smart contracts are also executed on the Ethereum Virtual Machine (EVM). **This provides compatibility to the**

vast resources of smart contract development with the Solidity language. Interactions with the smart contract for configuration changes and contract agreement are cryptographically signed to provide non-repudiation.

- **Native support for private transactions:** Confidentiality of transactions can be ensured by using the optional private channels. They are implemented by combining the default transaction model of Ethereum with an additional transaction manager (*Tessera*), which delivers encrypted messages to the specified recipients. **Only the hash value of the encrypted data is stored persistent and publicly on the Blockchain to ensure integrity and immutability.**
- **Modular architecture:** The project follows a very modular design and offers a custom plugin-system. **This allows simple selection of new consensus algorithms and modifications to the transaction manger.** All security critical operations are already concentrated into an independent module called *Enclave*. System-wide integration of TPMs for encryption and the Blockchain Security 2Go starter kit for secure key management is enabled by this architecture.
- **Privacy**: Although the previously described permission system can be used to limit the access to the network, the user identity is still concealed behind pseudonyms. For the DataVaults platform, this mechanism will be enhanced by the proposed DAA scheme (Section 4.2.3). **The personal app needs to provide a functionality to create a new pseudonym on demand (or automatically for new data sets) and register this address with the platform over a DAA channel.** The platform would store the new address in the list of authenticated users without gaining any additional knowledge about the user's identity.

The previously stated properties describe the fundamental role of the Blockchain regarding data transferred with smart contracts. However, **the encryption (e.g. ABE) and transfer of the main user-data to the cloud platform is mostly independent from the Blockchain and will be implemented with a dedicated transport channel. The processing of data to apply the anonymization techniques and provide further analytics services, is also done in separate modules.**

Furthermore, Quorum supports programmable Smart Contracts. With programmable, we mean that the programmer can define, using code, the functionalities that this Smart Contract will have. Some languages allow to program more functionalities than other, which are more restricted. In Datavaults, Smart Contracts – as described in Section 3 - in the private ledger will be used to process information about the access policies so they should permit storing basic data structures and permit read/write operations over them. Quorum also supports this functionality. Moreover, **Oracles might need to be used, although this needs to be studied carefully because they can overcomplicate excessively the final design of the solution.** Oracles are a special mechanism to communicate the Blockchain with external non-deterministic sources, so thanks to the Oracles, a Blockchain can get information from another source and use it to perform its calculations. In other words, Oracles feed the Smart Contract with external information that can trigger predefined actions of the Smart Contract. However, it is important to note that a Smart Contract does not wait for the data from the outside, instead, he needs to be invoked. Also, the main challenge with oracles is that people need to trust these outside sources of information.

4.4 SECURE TRADING MECHANISMS

As an advanced data sharing and privacy management framework, DataVaults framework and Blockchain-backed protocols allow various forms of corporate and user data to be **monetized and exchanged between different parties**. Individuals can earn **rewards for sharing data** on a secure, decentralized network; *rewards can either be monetization tokens, crypto currencies or other traditional financial services* (Section 5).

These **trading services** are enabled through the use of **smart contracts** (Section 3) **and trusted blockchain wallets** (Section 4.1) for securely storing and accessing all necessary user credentials for such trading deals. DataVaults leverages trusted computing technologies for delivering trust and payment services. **To eliminate impersonation and minimize transaction fraud, crptographic trust anchors will be embedded into a user's account wallet and trusted Blockchain control services will be designed, for enabling payer/payee authentication.** A user will receive payment if a data trading is successfully concealed and a payment event is triggered (note this is ensured through smart contracts). Payment transactions will be further recorded onto the ledger for validation (Figure 1).

DataVaults further specifies **advanced validation mechanisms** for guaranteeing the **correct execution of smart contracts** and the **prevention of possible entity misbehaviour** (data provider, broker and collector) in an attempt to violate user privacy and data security. This will be achieved through auditing user activity (potentially suspicious activity) records and specifying reward and "punishment" mechanisms as further incentivization. DataVaults will provide the following validation mechanisms customized to each system entity:

- **Validation of Data Provider.** In DataVaults, a data provider is offered three levels of validation to protect its execution rights from being infringed by a "*malicious*" broker. **Data storage validation** for verifying if data is stored correctly, in the data cloud market, and the corresponding metadata have been published on the ledger, following the agreed data management policies and the provider's preferred privacy level. **Block validation** for verifying if a block generated by the broker is valid. The TPM associated to the data provider will also help efficiently validate information stored on the ledger. **Payment validation** for checking if the payment amount is the one agreed in a trading smart contract.
- **Validation of DataVaults Cloud Platform (acting as the Data Broker)**. To maintain the **robustness** and **immutability** of the public ledger, DataVaults will register all data sharing and trading transactions (and their validation outputs) to the corresponding ledger blocks for further verification. The platform itself is also responsible for checking the outputs given by Data Providers and Data Seekers: in particular, the **data sharing and collection** processes must be aligned with the selected user preferences as well as the specified GDPR-based policies, the **formation of metadata** must follow the template provided by the broker, and **payment** must comply to the agreed trading smart contract.
- **Validation of Data Seeker.** This mainly reflects the validation of the **data collection** process in checking whether the **collected data falls under the correct category** based on the previously identified metadata (indicating whether the data broker commits a fraud during the data trading). For instance, if the collected data is categorized as weather information but the description of metadata (related to the collected data) is related to food information, then the data broker should be reported and given a (financial) penalty.

5 DATAVAULTS COMPENSATION MECHANISMS

5.1 DATA MARKETPLACES MAIN AXES

The following sections include an initial analysis of literature and of online resource on existing or past data marketplaces, to better understand the characteristics of those and acquire some deeper knowledge regarding the different **compensation mechanisms** and the way to **generate value for the different stakeholders that will conduct operations via the DataVaults platform.**

5.1.1 Data Marketplaces Categorisation and Characteristics

The exponential increase in the amount of available data along with new possibilities brought by data analytics and machine learning have made data the new oil of the 21st century [65].

Data that have been created, collected and used by individuals or organisations, can be sold to organisations to facilitate business processes and strategic targets, while their management raises costs, like any other material resource [66]. This has led to the emergence of new electronic markets that bring together data suppliers and data buyers and facilitate the trading of data as a commodity; the so-called data marketplaces [67]. Markets are the places where the interaction between buyers and sellers determines the price and amount of the exchangeable good [68], while marketplaces are the places where the preparation and execution of the actual transactions by the participating actors takes place. This means practically that data marketplaces are software infrastructures providing the appropriate frontend and/or backend interfaces that connect data buyers and sellers in order to allow them buy and sell data respectively [69].

Despite the extended research in the field, reaching a consensus to a uniform definition of data marketplaces remains a challenge, as the numerous designs from academia and industry demonstrate high diversity in the underlying business models, offered functionalities and other aspects. According to [70], several criteria allow for the characterisation of an electronic marketplace as “data marketplace”. Firstly, the provision of data and/or related services must be the marketplace provider’s primary business model. Secondly, the marketplace provider shall provide the users with an infrastructure to upload, browse, download, buy and sell machine readable data that is hosted in the provider’s infrastructure. These criteria lead to the exclusion of organisations offering open data, such as governmental organisations or NGOs, as trading of data is not their core business services. Furthermore, organisations offering links to data collections, without hosting data themselves are also excluded from the narrow definition. [71] have gone one step further, by enlisting in the definition of a data marketplace, not only the provision of an infrastructure enabling a seller to offer data in exchange for another valuable asset by the buyer, but also the implementation of data evaluation and validation, and incentivation mechanisms ensuring fairness and honesty between the trading actors.

The data marketplace landscape covers a diversity of data types, types of exchanges and use cases. From a business model perspective, [72] have identified the dimensions of a data marketplace, as deriving from the four basic elements of a business model (value proposition, value delivery, value creation and value capture) Figure 21.

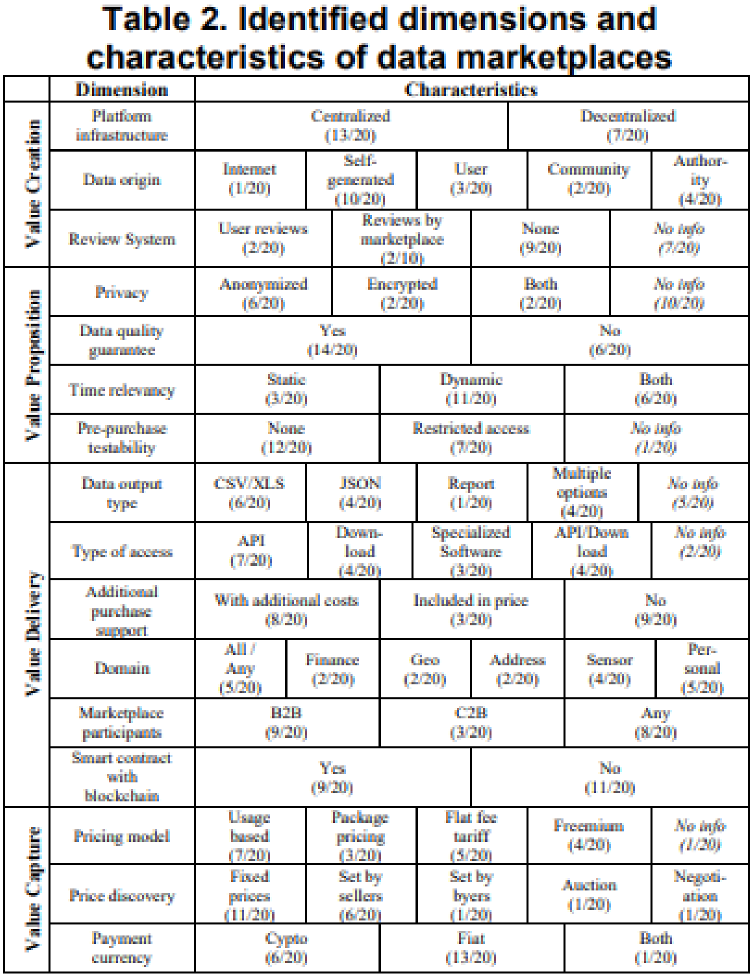


Table 2. Identified dimensions and characteristics of data marketplaces

	Dimension	Characteristics					
Value Creation	Platform infrastructure	Centralized (13/20)	Decentralized (7/20)				
	Data origin	Internet (1/20)	Self-generated (10/20)	User (3/20)	Community (2/20)	Author-ity (4/20)	
	Review System	User reviews (2/20)	Reviews by marketplace (2/10)	None (9/20)	*No info (7/20)*		
Value Proposition	Privacy	Anonymized (6/20)	Encrypted (2/20)	Both (2/20)	*No info (10/20)*		
	Data quality guarantee	Yes (14/20)	No (6/20)				
	Time relevancy	Static (3/20)	Dynamic (11/20)	Both (6/20)			
	Pre-purchase testability	None (12/20)	Restricted access (7/20)	*No info (1/20)*			
Value Delivery	Data output type	CSV/XLS (6/20)	JSON (4/20)	Report (1/20)	Multiple options (4/20)	*No info (5/20)*	
	Type of access	API (7/20)	Down-load (4/20)	Specialized Software (3/20)	API/Down load (4/20)	*No info (2/20)*	
	Additional purchase support	With additional costs (8/20)	Included in price (3/20)	No (9/20)			
	Domain	All / Any (5/20)	Finance (2/20)	Geo (2/20)	Address (2/20)	Sensor (4/20)	Per-sonal (5/20)
	Marketplace participants	B2B (9/20)	C2B (3/20)	Any (8/20)			
	Smart contract with blockchain	Yes (9/20)	No (11/20)				
Value Capture	Pricing model	Usage based (7/20)	Package pricing (3/20)	Flat fee tariff (5/20)	Freemium (4/20)	*No info (1/20)*	
	Price discovery	Fixed prices (11/20)	Set by sellers (6/20)	Set by byers (1/20)	Auction (1/20)	Negoti-ation (1/20)	
	Payment currency	Cypto (6/20)	Fiat (13/20)	Both (1/20)			

Figure 21: Identified dimensions and characteristics of data marketplaces [73]

The identified dimensions are presented in more detail below:

- Platform Infrastructure: This aspect is related to the architecture of the data marketplace. It can be centralised, meaning that data are offered from a centralised location, or decentralised (for example through a blockchain network), wherein data remain at the data provider's side. The study found that approximately two thirds of investigated data marketplaces are centralised.
- Data Origin: It specifies the source of the exchanged data, as Internet-generated, self-generated, user-generated, community-generated, government/authority-generated data. The majority of data marketplaces handle self-generated data from private sources.
- Review System: Whether the data assets can undergo an evaluation process either by the users or the marketplace itself. In sixteen out of twenty marketplaces however, no review mechanism is foreseen, or related information is not provided.

- Privacy: This dimension pertains to the design and provision of privacy-preserving mechanisms to protect users' privacy and confidentiality. Encryption or anonymisation techniques are employed by half of the marketplaces towards this purpose.
- Data Quality Guarantee: Some data marketplaces offer guarantees about the quality of the offered data assets.
- Time Relevancy: It refers to the dynamic aspect of the offered data assets, meaning whether it needs to be regularly updated to remain valid, or it remains unchanged after its creation.
- Pre-purchase Testability: Some marketplaces offer previews of the data assets under review for purchase by the data seekers, to see if they match their needs, in the form of complete or restricted access. However, this could jeopardise the privacy of data providers.
- Data Output Type: The purchased data assets can be exported from the data marketplace in one or more of the following formats: as semi-structured data (e.g. JSON), in tabular format (e.g. CSV/XLS), in visualised formats (e.g. PDF, DOC, JPEG).
- Type of Access: The users can access the data assets either through provided interfaces (APIs), as downloadable files, or with the use of specific software. Some data marketplaces offer a multitude of access mechanisms.
- Additional Purchase Support: It entails the provision of additional services to the trading of data, such as data analytics. These services are offered for free or with an extra charge.
- Domain: This dimension refers to the actual information contained by the exchanged data assets, as for example personal data, geolocation data, financial data and more.
- Marketplace Participants: The data trading actors involved in the data trading can be individuals (clients) or businesses. From the investigated data marketplaces, either there was no specific focus, or it was mainly on business-to-business (B2B) interactions, and only a minority was oriented at the client-to-business (C2B) model.
- Smart contract with Blockchain: The use of smart contracts as a privacy and safe payment enabler has been adopted by half of the data marketplaces for the enforcement of trust in transactions.
- Pricing Model: It refers to the marketplace's strategy to profit from its business activities. Usage-based models (e.g. based on API calls number, or time), package pricing (i.e. fixed price for an amount of data), flat fee tariffs (i.e. recurring fee to provide total access) and freemium models (i.e. basic features offered for free and advanced features offered for a fee) are among the most widely adopted pricing models in the field.
- Price Discovery: The exchangeable data assets' prices before the transaction are in their majority determined either based on fixed prices or are set by data sellers. In a few examples, data prices are set by the buyers and through biding or negotiation processes.
- Payment Currency: The data marketplaces handle payments with specific currencies. Fiat currency is the prevailing type of payment, although there exist marketplaces offering cryptocurrencies or both.

After exploring the similarities among the investigated data markets and combining different dimensions, the authors [72] came up with a data marketplace taxonomy consisting of four fundamental data marketplace archetypes Table 3, namely:

- Centralised data trading – data marketplaces where data is hosted in a single point
- Centralised data trading with smart contracts – centralised data marketplaces incorporating smart contracts for the transactions to ensure privacy and trust
- De-centralised data trading – data marketplaces that rely on decentralised infrastructures such as blockchain, ensuring data quality-
- Personal data trading – data marketplaces enabling individuals to sell their data through dedicated software, following the C2B model.

Data Marketplace Archetype	Centralised Data Trading	Centralised Data Trading with Smart Contract	De-centralised Data Trading	Personal Data Trading
Data Marketplace	Quandl	Dawex	IOTA	Datacoup
Value Creation	Centralised	Centralised	De-centralised	De-centralised
Value Proposition	Anonymised Dynamic Datasets	Encrypted Static and Dynamic Datasets	Encrypted Dynamic Datasets	Anonymised Dynamic Datasets
Value Delivery	• API or download • Restricted access to data samples • B2B • No smart Contract	• API or download • Restricted access to data samples • B2B • Smart Contract	• API • No test data samples • B2B • Smart Contract	• Specialised software to accessNo test data samples • C2B • Smart Contract
Value Capture	• Freemium pricing • Prices set by sellers • Fiat currency	• Usage based pricing • Prices set by sellers • Fiat currency	• Flat fee pricing • Prices set by sellers • Crypto currency	• Usage based pricing • Fixed prices • Crypto currency

Table 3: Illustrative examples of data marketplace business model archetypes [72]

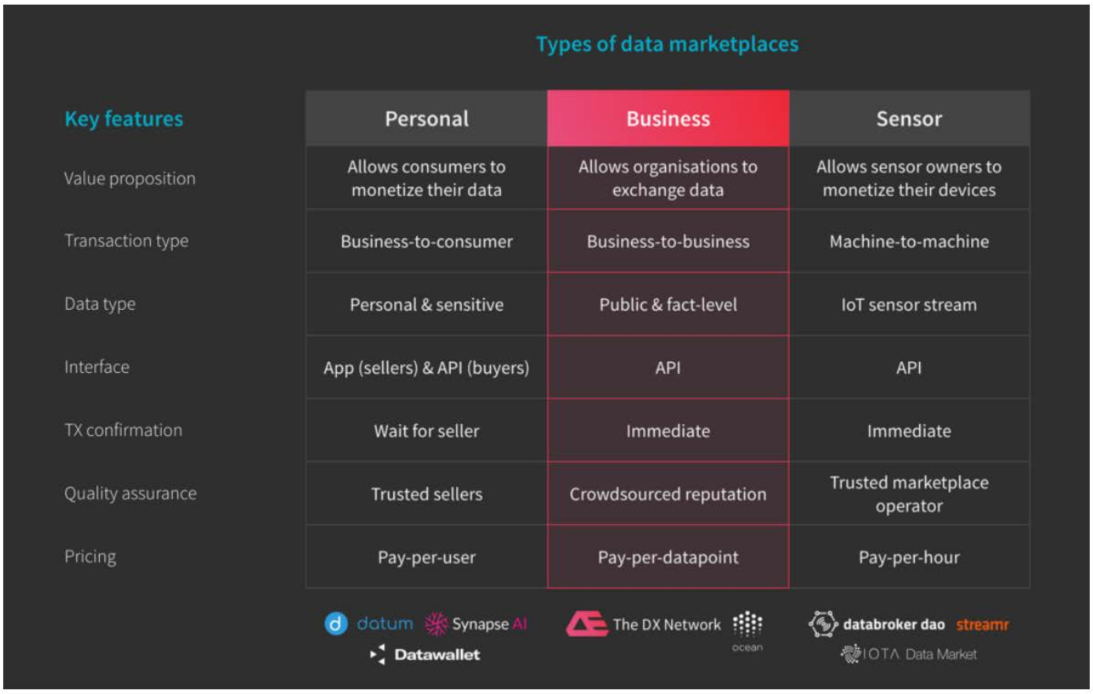
Another intuitive way to categorise data marketplaces is based on the data type – personal, business or sensor data- and has been presented in [69].

In personal data marketplaces such as Datum and DataWallet, the focus is on B2C transactions, the users interact through dedicated software (apps for sellers and APIs for buyers), to exchange personal and sensitive data spanning from email addresses to fitness tracker measurements.

In business-oriented data marketplaces, organisations exchange enterprise knowledge usually in the form of structured and big data.

Finally, sensor data marketplaces are dedicated to the collection and trade of real time data coming from sensors that measure pollution, traffic and more.

The three identified categories have applied different pricing, quality assurance and transaction methods to fit the needs of the participating actors as well as meet the security and technical aspects that arise from the nature of the traded data.



Types of data marketplaces

Key features	Personal	Business	Sensor
Value proposition	Allows consumers to monetize their data	Allows organisations to exchange data	Allows sensor owners to monetize their devices
Transaction type	Business-to-consumer	Business-to-business	Machine-to-machine
Data type	Personal & sensitive	Public & fact-level	IoT sensor stream
Interface	App (sellers) & API (buyers)	API	API
TX confirmation	Wait for seller	Immediate	Immediate
Quality assurance	Trusted sellers	Crowdsourced reputation	Trusted marketplace operator
Pricing	Pay-per-user	Pay-per-datapoint	Pay-per-hour

Some of the key properties of blockchain-powered data marketplaces categorized by data type [The DX Network]

Figure 22: Types of data marketplaces [69]

5.1.2 Data Marketplace Actors

Most data marketplaces in literature involve mainly three main actors [74, 75, 76, 77, 78]:

- the **data providers** (referred to also as supplier, owner or simply individual) that provide their data for the appropriate monetary compensations,
- the **data seekers** (also found as buyers or data consumers) that are willing to pay a certain amount of money to acquire data assets, and
- an **intermediary** (market maker, broker, notary, orchestrator) that acts as a trusted mediator between the buyers and sellers

The role of the mediator is the provision of an infrastructure that allows the execution of all processes pertaining to data upload and collection, storage, querying, pricing, data and money transactions and finally data provision through APIs or downloadable formats [79]. However, depending on the specific marketplace's design, the intermediary may also be responsible for the perturbation of data assets in order to ensure privacy protection [77], the pre-processing of collected raw data, or may be attributed the role of an authority that dominates the market. This authority witnesses and validates the various transactions that take place and enforces the marketplace's policies [79], calculates data prices based on demand and supply but also on the nature of the data, or even resolves conflicts between trading parties.

In most cases, the data marketplace acts in the role of intermediary, although there are examples of decentralised architectures where the role of the mediator can be undertaken by any qualified participant entering the market as in the case of the data seeker and supplier role, or is even bypassed through the appropriate deployment of blockchain and smart contracts. It should be noted that data marketplaces are not always independent, as they could be operated by the same actors involved in the supply of data, a usual phenomenon with large companies, in contrast to smaller companies that employ third-party marketplaces that are neutral as do not stand by either side of the transaction.

5.1.3 Data Marketplace Architecture

Data Marketplace frameworks usually incorporate the following mechanisms, as described in [69]:

- A Data Storage module responsible for the collection and storage of data either in centralised or decentralised infrastructures,
- A Querying System to enable data seekers find data assets that match their criteria,
- A Pricing Mechanism that allows the definition of a price for a quantity of data prior to their purchase based on the pricing scheme in effect,
- A Payment System for the execution of the monetary transactions between the trading parties,
- An Incentive System providing mechanisms that promote honest behaviour from buyers and sellers and ensure the trustworthiness of data.
- Decentralized user applications (dApps) may also be provided by decentralised blockchain-based marketplaces to support users during the overall data exchange process.

For the design and implementation of each component, one shall not only consider the core functionality to be offered, but also any accompanying features that will ensure the key principles for creating a reasonable and fair personal data marketplace are met. Several aspects that could jeopardise the balance and fairness of the data market, as well as the ownership and privacy rights of users have been highlighted in literature. Indicatively, we will refer here to some of the most discussed challenges.

The first challenge pertains to the definition of an appropriate price by the sellers, that will maximise their gains while making their data affordable to buyers. It is often the case for massive amounts of data to remain on the shelf because of overpricing [72]. A recommendation mechanism by the data marketplace to make price suggestions to the sellers, is often incorporated in order to help in price selection.

Another complication comes from the asymmetry of access to the data assets prior to their purchase. Malicious actors acting as sellers could take advantage of this situation and sell junk data to buyers that cannot verify their validity, while on the other hand malicious actors acting as buyers could exploit any information leakage for data testing prior to purchase [75].

Another interesting point is the unintended inference of sensitive information acquired by data buyers. Such risks affect heavily IoT data, firstly because of their inherently personal and intimate nature, as well as because of the data mining and metadata generation techniques that could reveal intimate details not explicitly contained in the datasets [75].

A risk for privacy and ownership is imposed also by a situation called arbitrage. This is the ability of buyers to combine the results of multiple queries and derive the answer for a new query without paying the full price.

The dependency and possible bias in the results of a data marketplace, in favour of selected data suppliers, has also led to the design of appropriate mechanisms, with a focus on decentralised systems, to avoid any accusations of "bribing" [69].

Security issues can stem from technical implementations, as for example in, where transactions that are registered off-chain to reduce computation time and costs raise the need for additional protection, or in who highlight the problem of possible linkage of user information to blockchain transactions.

The use of public ledgers for the registration of transactions creates discussions around user privacy and has led to the implementation of novel protocols, such as the Masked Authenticated Messaging (MAM), aiming to facilitate privacy and integrity at the same time.

[74] have compiled a list of five distinctive features that will address challenges as the above and will result in designing a fair personal data marketplace. These features are:

- ensuring privacy protection,
- allowing querying over the collected data,
- applying an arbitrage-free pricing model,
- encouraging truthful data sharing and finally,
- providing unbiased results.

These principles raise business and technical requirements that will be considered for the design of DataVaults data marketplace in the respective sections.

5.2 Economics of Personal Data

Personal data has value to both the Individual who owns it and to government services and companies who would like to acquire and analyse it.

Currently there are many Internet giants who follow a simple formula to acquire personal data. They offer a free service, attract Individuals who provide their personal data, and then monetize the personal data by selling it, or by selling information derived from it, to third parties. In turn, many Individuals are willing to provide their personal data in return for access to online services and social networks. But as Individuals become more aware of the use of their data by corporate entities, of the potential consequences of disclosure, and of the ultimate value of their personal data, there has been a drive to compensate them directly.

Assuming that an Individual wants to take control of his/her own personal data and demand direct compensation for it, the following two paths have been established and explored by start-ups around the world: first, a platform could aid the Individual to create personal data vaults. The platform buys the raw personal data from each Individual and compensates them accordingly. Then, the platform allows data consumers to search for personal data and acquire them for a price set by the platform. The platform guarantees the correct functioning of the market. A negative of this case is that some Individuals are not convinced to sell their raw data. A second option could be that an application aiding the Individual to collect his/her personal data through a mobile application and store all of his/her personal data in his/her device. Then, data consumers should contact each Individual and ask for a fair price to acquire his/her personal data. A negative of this case is that it might be inefficient and impractical for data consumers to buy individual raw data one at a time.

In order for an Individual to be convinced to participate in such data markets, it is not only money that plays a role. As it has been identified, Individuals in general value the following:

- by whom and how the data will be used;
- sensitivity;
- future risks/impacts; and
- money.

It can be understood that there is a fundamental clash between sharing personal data and getting compensated about it and at the same time keeping the perceived privacy loss of an Individual at the desired level. The interests of Individuals and government services and companies with respect to personal data are often at odds and a rich literature on privacy-preserving data publishing techniques has tried to devise technical methods for negotiating these competing interests.

In summary, the proposed data markets work as follows. Individuals provide personal data to the data market based on their privacy preferences and receive appropriate compensation. The data provided by the Individual is stored by the trusted data market. To make a profit, the data market needs to charge the data consumer a certain fee, and the data consumer can purchase data products based on their willingness to pay [74].

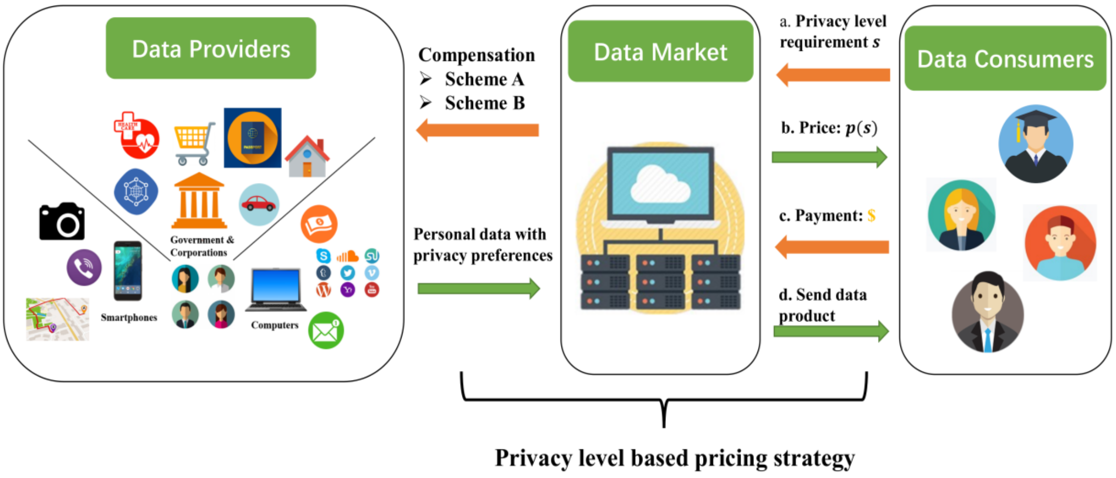


Figure 23: A typical personal data market framework [74]

The main challenges to be resolved are:

How to provide a fair-trading mechanism between data providers and data platforms?

Much like traditional commodity trading, the most important focus in the data market is fairness and truth. Of course, this is a basic requirement for all trading processes. Under a linear compensation scheme, most Individuals will always set very high price for maximum benefit. One problem is that shrewd Individuals could tamper personal data and send personal data of low value to the platform, in order to maximise their profit without equal privacy loss. A second problem is that if a linear compensation scheme it is employed, it means sellers offering different levels of sensitivity receive the same amount of compensation, which is unfair to Individuals who provide highly sensitive information. Furthermore, it will accelerate the loss of Individuals who offer the stable and high-quality data sources to the data platform, which is not conducive to the long-term stable development of the platform. To establish a fair privacy compensation mechanism, the platform should encourage real privacy assessments, and without compromising the interests of highly sensitive information providers, the platform should provide appropriate compensation schemes that correspond to the data provider's privacy attitude [74].

What is the Individual's attitude toward privacy data?

Each Individual's privacy attitude is different, so the utility of publishing data is different. Personal data collected vary in type and quality and their monetary value should also vary depending on the need and the requirements of the data consumers. The platform should propose a different compensation schema for personal data of different utility; in [74] it is argued that a number of five utility levels is the optimal number in order to accommodate all possible types of Individual's attitude towards privacy data.

Utility levels are linked to loss of privacy. The greater the loss of privacy, the more information is disclosed, and thus the utility of the data is higher. For those Individuals sharing highly sensitive personal data ("Risk Taker"), moderate privacy losses should be linked to a small amount of compensation, but for significant privacy losses, a huge amount of compensation should be required. For those Individuals sharing moderate or low sensitive personal data ("Risk Averse"), even the smallest loss of privacy has a non-zero small compensation. However, even the largest compensation is far less than the "Risk Taker's" compensation.

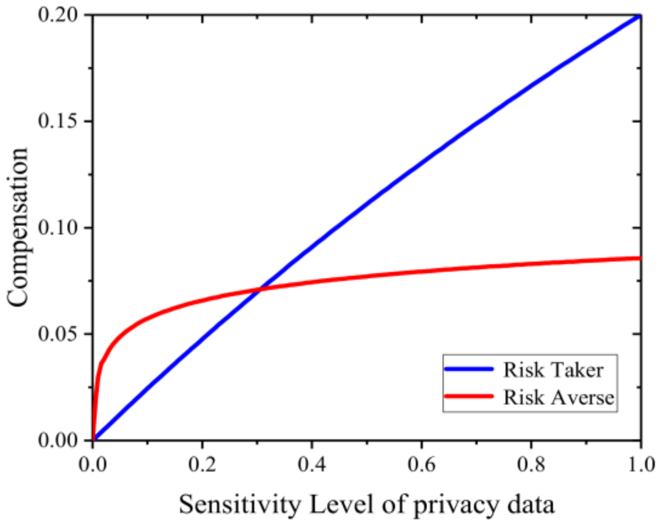


Figure 24: Compensation mechanism for two different privacy attitudes [74]

How to price personal data?

For the Individual, a payment scheme is a non-decreasing function representing a promise between a market maker and a Individual on how much Individual should be compensated for their actual privacy loss. As we saw in Figure 24 above, it should be correlated to the attitude of the Individual towards privacy. This logic has been expressed both in [74] and in [75].

For the data consumer, willingness to pay is a function which combines the type of heterogeneous consumer and the sensitivity level of privacy data that the consumer wants to purchase. There is an infimum point which indicates that the sensitivity level of the data is too low, and cannot bring useful value to consumers, so consumers judge the value of such data as 0. As the level of data sensitivity increases, the resulting utility will also increase, which will increase the willingness of consumers to pay. Below a certain point, as the sensitivity level increases further, the increase in data utility will decrease [75].

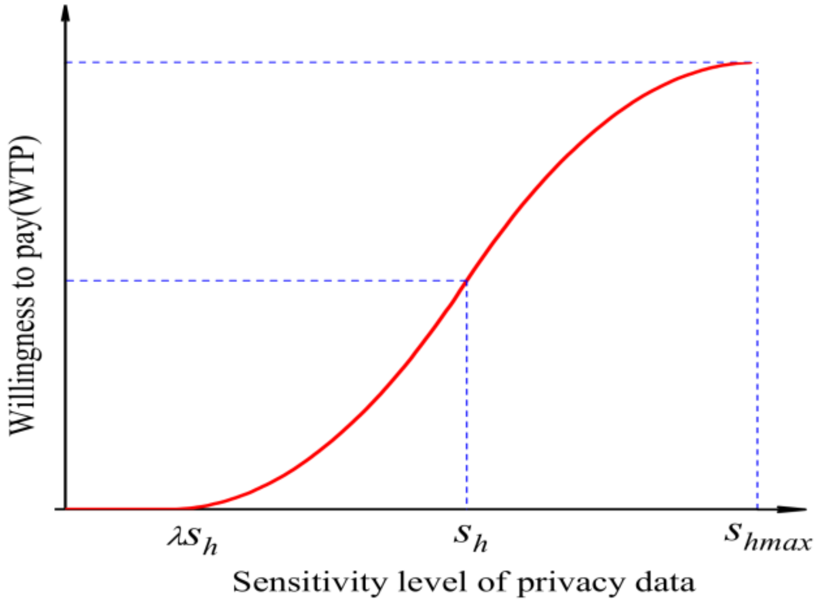


Figure 25: Consumer's willingness to pay for privacy levels [75]

Regarding the broker's net payoff, this is calculated by subtracting the total costs of collecting and selling the data from the total revenue. The cost of data for the broker can be classified into two categories:

- The first is the cost of purchasing the data from providers, which is calculated by multiplying the quantity of data purchased by the price per data unit.

The second is the cost of collecting the data, which includes infrastructures, staff and departments, licenses, advertisements, distributions, compensations, etc.

5.2.1 Rewarding Mechanisms of Data Marketplaces

How is the value of personal data unlocked and translated in actual revenue for the Data Providers? The most common motivation mechanisms that are employed by data marketplaces to attract Individuals into sharing their personal information can be largely categorised under two groups: monetary and nonmonetary rewards.

Monetary rewards: The Providers receive for data they share with Data Seekers a certain monetary value, that has been agreed upon prior to the data exchange. Both Fiat money and Cryptocurrencies are used as means of monetary payments, depending on the data sharing infrastructure. The price of the data asset can be set by the Provider, can be regulated by the Data Marketplace, can be proposed by the Data Seeker, and combinations of the above. The majority of the identified data marketplaces employ monetary rewarding mechanisms for the compensation of the Individuals. Citizen Me [3] is a paid mini survey application that compensates users with cash through their connected PayPal account, straight after they have completed a survey. Datum[4] is a blockchain-enabled data storage and monetisation platform where users are rewarded with DAT tokens paid out as a fixed percentage for each transaction of their data from buyers. The crypto-powered protocol provided by Fysical [5] enables consumers to be compensated for any data purchases with ERC20 Tokens. Wibson also provides a decentralised platform for Individuals to securely share and monetise on their data with ERC20 Tokens that are transferred to them in batch payments. In business-to-business data marketplaces, monetary compensations are also widely used, as for example, in Quandl[6], Dawex[7], DX Network[8] and Ocean[9] where data flows are commercialised and turned in revenue streams with the use of Fiat or cryptocurrency.

Nonmonetary rewards: Rewards for data sharing, other than money, can be issued in the form of royalties, permissions, badges, reputation or lottery tickets for a network reward function, and more. The Panel App[10] incentivises users into sharing location-based survey data by accrediting points for each survey, that can be used among others for prizes, rewards and gift cards. Personalisation of services has been used as a reward by the people.io[11], who envisioned to show users the benefits of having power over their data through enabling data seekers provide data-centric services. Furthermore, people.io offered credits to users for every completed survey, that could be redeemed for gift cards or be donated[12].

Hybrid: Numerous personal data marketplaces have adopted a hybrid approach, by pffering both monetary and nonmonetary compensations to their users. MyDataMood[13] is a new initiative that grants users with access to a loyalty club, the Mooders Club, once they have

[3] https://www.citizenme.com/public/wp/for-citizens/
[4] https://datum.org/
[5] https://fysical.org/
[6] https://www.quandl.com/
[7] https://www.dawex.com/en/data-exchange-platform/#unlockValue
[8] https://dx.network/
[9] https://oceanprotocol.com/technology/roadmap
[10] https://play.google.com/store/apps/details?id=com.sewichi.client.panel&hl=en
[11] http://people.io/about.html
[12] https://lovelymobile.news/telefonica-partnership-gives-consumers-more-control-over-personal-data/
[13] https://www.mydatamood.es/faq

shared personal information with three organisations. Once becoming a member, the Individual can enjoy benefits such as direct discounts to well-known brands. Furthermore, the Individual receives direct economic award through their Mooders account, whenever a Data Seeker buys their data. The Trusts[14] EU funded project also aims to include both methods of awarding in their personal data marketplace. Datacoup[15] was another platform bringing control and value back to Individuals, in the form of cash, cryptocurrencies and discounts. UBDI[16] is a platform enabling research by making personal data available for studies, at the appropriate fee. This is paid to Individuals both in cash and in UBDI points through Paypal's Hyperwallet. UBDI points are used to provide users with special rewards, and can be transferred to other systems, such as a bank, Paypal or Venmo account. BitsAboutMe[17] offers Individuals monetary rewards that are paid directly in a bank account, while it facilitates the provision of special offers by trusted partners through its marketplace.

5.3 MONETISATION ENABLING TECHNOLOGIES (TECHNOLOGY AXES)

5.3.1 Current State of Micropayments using DLTs

The following sections provide an overview of the current state of the art regarding micropayments using distributed ledger architectures, exposing the main problems and challenges faced, aiming at understanding the main features and challenges that the technical implementation of DataVaults should include and the challenges it might face.

5.3.1.1 Blockchain protocol and problems with micropayments

Micropayments come in the category of electronic payment systems, which are financial transactions that take place through an electronic medium without using paper checks or cash. A micropayment is a financial transaction involving an amount of money up to few euros or even a fraction of a cent.

Two problems arising from the use of blockchain protocol in order to implement micropayments is network scalability and transaction costs.

The first problem stems from the nature of blockchain protocol which by definition dictates that all state modifications to the ledger are broadcasted to all participants. It is through this consensus of the state that everyone's balance is agreed upon. If each node in the bitcoin network must know about every single transaction that occurs globally, that may create a significant drag on the ability of the network to encompass all global financial transactions.

The payment network Visa can achieve 47,000 peak transactions per second (tps) while Bitcoin supports less than 7 transactions per second with a 1 MB block limit. Clearly, achieving Visa-like capacity on the Bitcoin network isn't feasible today. No home computer in the world can operate with that kind of bandwidth and storage.

[14] https://www.technologyreview.com/2020/08/11/1006555/eu-data-trust-trusts-project-privacy-policy-opinion/

[15] https://datacoup.com/#

[16] https://www.ubdi.com/individuals/how-earning-works

[17] https://bitsabout.me/en/

The second problem is the relatively large transaction fee. The bitcoin transaction fee is a payment to the miners, the typical fee ranges from approx. €0.1 to €0.25. If the micropayments are of similar sizes, the added fee could represent a high percentage of the payment itself.

5.3.1.2 Aggregating payments off-chain

The most common approach to address these issues is an aggregation, which is supported by the following conclusions. Payment aggregation replaces many micropayments with a small number of total payments to be recorded in the ledger. With the aggregate, transactional payments (fees) are paid only for such consummated transactions. In other words, aggregation reduces not only the number of entries but also the transactional costs per payment. There are two types of aggregation in centralized systems.

1) accurate; for instance, all phone calls are accounted for, but paid as a lump sum once a month, and
2) probabilistic.

By deferring telling the entire world about every transaction, doing net settlement of their relationship at a later date enables Bitcoin users to conduct many transactions without bloating up the blockchain. Then comes the need for a trusted custodian where transactions are offloaded. The custodian is a trusting third party who holds one’s coins and updates balances with other parties. Trusting third parties to hold all of one’s funds creates counterparty risk and transaction costs.

The following technical solutions are three proposals currently on the table to solve the problem of aggregating payments off-chain and only publishing them as aggregated records, without the need of a third-party custodian (middleman).

Network of micropayment channels (c-lightning project) Joseph Poon and Thaddeus Dryja propose a decentralized system whereby transactions are sent over a network of micropayment channels (a.k.a. payment channels or transaction channels) whose transfer of value occurs off-blockchain. Using a network of these micropayment channels, Bitcoin can scale to billions of transactions per day with the computational power available on a modern desktop computer today. Sending many payments inside a given micropayment channel enables one to send large amounts of funds to another party in a decentralized manner]. It is known by the name “c-lightning project”.

To achieve fast and cheap micropayments, smart contracts are used. Via a network of multi-signature transactions, any participant on the Lightning Network can pay someone else. This is done through a two-party consensus, known as a payment channel.

Even though two parties are involved, a person doesn’t need to open new payment channels with every new party they want to transact with. For instance, Alice may not have an open channel with Charlie, but Alice is indirectly connected with Charlie through Bob. With the Lightning Network, anyone can transact with someone else who is connected to their network of payment channels. In theory, everyone should be connected with others on the network through a small number of nodes. To boost the number of people using the network, Lightning is incentivizing LN adopters to run connecting nodes by enabling them to collect small fees each time a transaction is conducted through their connections.

With every transaction conducted, both parties must agree on the new balance to maintain a clear record as to who owns what bitcoin stored in the multisignature wallet. When one wants to update the balance with a new balance, both parties must consent to the new balance.

Rather than conducting their business via the public blockchain, the Lightning Network's use of payment channels enables users to handle their business directly with each other. This means users can avoid expensive and time-consuming interactions with the blockchain, particularly if it involves micropayments. It's only when both parties want to terminate the channel or if there is a dispute that they fall back to the most recent balance sheet provided by both parties, which will determine how the funds in the multisignature wallet are split up. This is then conducted on-chain to provide a record of the transaction.

Use "lottery tickets" instead of payments (Randpay) Oleksii Konashevych and Oleg Khovayko try to move micropayments off-chain while at the same time excluding trusted third parties; the idea is to provide users with capabilities to interact with each other peer-to-peer, and at the same time, not to use an existing approach for peer-to-peer protocols that require the creation of so-called "payment channels" because they typically require also performing opening and closing blockchain transactions, while the aim is to reduce them. It is known by the name "Randpay".

The essence of the idea is to finalize each settlement, not with a payment but with a "lottery ticket". Only the winning "lottery tickets" of the service provider (Bob) will be published into the blockchain as the transaction. Bob (the service provider) provides Alice (the user of the service) a "lottery ticket", which carries the information of the space of payment addresses, where one is Bob's winning. Alice makes her random choice picking one address from the provided space, generates the raw transaction, and sends it directly to Bob. If Alice's choice contains the payment address to which Bob has the private key, Bob will sign the raw transaction and publish it on the blockchain and so he will take the money. If Alice has chosen a payment address to which Bob does not have a key, this transaction will not be published and just set aside, and Bob will deliver Alice the product for free [2020_ Randpay: The technology for blockchain micropayments and transactions which require recipient's consent].

In essence, the user of the service most of the times gets the service for free ("wins the lottery"). On the other hand, when the provider "wins the lottery", the user pays multiple times the cost of the service (i.e. the user makes calls for 100 minutes, ends up to pay 1 minute, but the price is 100 times higher than the usual price-per-minute). The number of transactions directly with blockchain is obviously reduced by 100 times in the example above; the inventors of Randpay concept argue that when regularly using a service which by nature is used many times and occurs minimal costs each time (e.g. phone calls), neither the user nor the service provider lose, as the compensation asymptotically reaches the classic "per minute" charge.

Utilise Trusted Execution Environments (Teechain) Lind, Eyal, Kelbert, Naor, Pietzuch and Sirer describe Teechain, an off-chain payment protocol that utilizes trusted execution environments (TEEs) to perform secure, efficient and scalable fund transfers on top of a blockchain, with asynchronous blockchain access. Teechain introduces secure payment chains to route payments across multiple payment channels. Teechain mitigates failures of TEEs with two strategies: (i) backups to persistent storage and (ii) a novel variant of chain-replication].

TEEs are a hardware security feature in which code and data in a trusted memory region are isolated and protected from the rest of the system. Because the TEEs protect the internal channel state and release it only upon channel termination, they ensure that users cannot launch attacks by using stale state. Intel SGX is included in modern CPU's as a set of instructions that increases the security of application code and data, giving them more protection from disclosure or modification.

The main idea behind Teechain is to aggregate all micropayments inside the CPU of the very machine the user is using, only to publish it at the blockchain network at the end of the series of transactions with the other party. Obviously, the number of interactions with blockchain is minimised, while at the same time TEE guarantees the security of the user's coins even in the event of a malicious person having physical access on the very machine s/he is using.

5.3.2 Challenges

Using cryptocurrency to pay for online purchases entails a new generation of threats related to the possible deanonymization of users with the help of information logged by activity trackers typically found on commercial websites. On most shopping websites, third party trackers receive information about user purchases, e.g. for purposes of advertising and analytics.

In [80], it is explained how third-party web trackers can deanonymize users of cryptocurrencies and two attack methodologies are laid out: in a first attack, if the user pays using a cryptocurrency, trackers typically possess enough information about the purchase to uniquely identify the transaction on the blockchain, link it to the user's cookie, and further to the user's real identity. A second attack shows that if the tracker is able to link two purchases of the same user to the blockchain in this manner, it can identify the user's entire cluster of addresses and transactions on the blockchain, even if the user employs blockchain anonymity techniques such as CoinJoin. For examples of these attacks see section I of [80].

Cryptocurrency anonymity is a new research topic, but it sits at the intersection of anonymous communication and data anonymization, both well-established fields. Unfortunately, it seems to inherit the worst of these two worlds. Sensitive anonymized data must be publicly and permanently stored, available to any adversary. De-anonymization attacks are passive and hence can be retroactively applied to past purchases. Moreover, privacy depends on subtle interactions arising from the behavior of users and applications.

Trying to devise defense methodologies on the abovementioned threats, it is observed that the first attack exploits the inherent tension between privacy and e-commerce, and the second attack exploits the inherent tension between privacy and the public nature of the blockchain. Thus, all mitigation strategies come with tradeoffs.

5.3.3 Technology Considerations for DataVaults

Based on the above, and having in mind the high level architecture of DataVaults, as well as the need to devise 2 different ledgers for accommodating public transactions (e.g. between data seekers and the DataVaults Cloud Platform) and private transactions (e.g. between the DataVaults Cloud Platform and Individuals (data owners), the consortium proceeded with an analysis of blockchain technologies which match the requirements imposed by the project

Based on the above, the following blockchain enabling technologies are considered for use within the DataVaults project, as those do cover the requirements of the overall concept that steam from the early architectural discussions conducted by the consortium to select a solution that can be at the same time performant, efficient and also serve the security and privacy requirements that are imposed by the different technologies and cryptographic primitives that should be used by the different DataVaults modules.

- **Ethereum** a global, decentralized platform for money that lets users send cryptocurrency to anyone for a small fee. On Ethereum, you can write code that controls money, and build new kinds of applications that everyone can use and no one can take down. It's the world's programmable blockchain.
  Ethereum builds on Bitcoin's innovation, with some big differences. Both allow the use digital money without payment providers or banks. But Ethereum is programmable, making it account for more than payments. It's a marketplace of financial services, games and apps that can't steal users’ data or censor them.
- **Quorum**, as a fork of Ethereum 1.0, is an open-source blockchain platform that combines the innovation of the public Ethereum community with enhancements to support enterprise needs. GoQuorum was originally developed by J.P. Morgan Chase. It is a fork of Go-Ethereum (also known as Geth), which is a mainnet Ethereum client developed by the Ethereum Foundation.
  ConsenSys Quorum enables enterprises to leverage Ethereum for their high-value blockchain applications. Businesses can rely on the Quorum open-source protocol layer and integrate on top of it product modules from ConsenSys, other companies, or your own in-house development team to build high-performance, customizable applications.

6 CONCLUSIONS

This final section will act as a synopsis of this deliverable and summarize its findings. The scope of this deliverable was to provide a detailed analysis of the Blockchain Distributed Ledgers, leveraged by DataVaults, towards the creation of a digital marketplace. This is achieved through the design and implementation of **policy-compliant Blockchain structures** to be enhanced with advanced **on- and off-chain data and knowledge management services** through the specification of appropriate security services including **access control, smart contract composition (reflecting the data sharing configurations defined by the Individuals), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of trust anchors) as well as privacy-preserving services.**

For the former, DataVaults is based on a hybrid **Blockchain-powered infrastructure** (integrating the use of both private and public ledgers) that will facilitate **sealing of smart contracts** on the side of the Individuals, as well as their compensation for assets that have been procured by Data Seekers. The secure data storage, publish and sharing will follow the latest trends in DLTs to rely on trust anchors of different types, each being important in terms of some dimension of **policy, technology, data, security, assurance and more.** DataVaults relies on a combination of advanced set of **cryptographic trust anchors towards binding entities and attributes to data subjects and data principals**, as well as to actors within the system that operate the DataVaults trust framework.

In this context, DataVaults Blockchain will mainly inherit the intrinsic functions from the **Quorum technology** (Section 4.3) to achieve the storage, publish and data sharing for all authenticated members, as well as data broker and outsiders who can first read the metadata on the public ledger before requesting access to any stored data. Different from current Blockchain functions, DataVaults will consider **secure onchain data searching so as to provide a privacy-preserving way for Data Seekers to search preferred information without leaking sensitive information of the data (on private ledger) before being granted read rights.**

For the latter, the deliverable also presented and assessed the cryptographic primitives and protocols leveraged by DataVaults for achieving enhanced user privacy protection - needed to secure different types of information, while still allowing advanced knowledge discovery through the provision of **enhanced data search services** (i.e., Searchable Encryption), and **advanced security and privacy-preserving primitives** (i.e., data anonymization and pseudonymization techniques) for authentication, authorization, attestation and verification through the use of trusted computing technologies. Such an analysis will serve as the basis (and provide valuable insights) on the definition of the overall DataVaults conceptual architecture and identification of all internal interfaces to be documented in D5.2 [16].

Finally, a detailed analysis on the existing types of compensation mechanisms that can be used in the context of DataVaults was performed for this deliverable. This will set the scene for the finalization of the workflow of actions towards secure and fair data trading in Deliverable D2.3.

7 REFERENCES

[1] W3C, «RDF 1.1 Concepts and Abstract Syntax,» [En línea]. Available: https://www.w3.org/TR/rdf11-concepts/.

[2] W3C, «Data Catalog Vocabulary (DCAT) - Version 2,» [En línea]. Available: https://www.w3.org/TR/vocab-dcat-2/. [Último acceso: 07 2020].

[3] L. M. Dan Brickley, «FOAF Vocabulary Specification 0.99,» 2014. [En línea]. Available: http://xmlns.com/foaf/spec/.

[4] W3C, «ODRL Information Model 2.2,» 2018. [En línea]. Available: https://www.w3.org/TR/2018/REC-odrl-model-20180215/.

[5] W3C, «Data Privacy Vocabulary v0.1,» 2019. [En línea]. Available: https://w3.org/ns/dpv.

[6] SIOC Project, «SIOC Core Ontology Specification,» 2018. [En línea]. Available: http://rdfs.org/sioc/spec/.

[7] D. Consortium, «D1.2 The DataVaults Core Semantic/Data Model,» 2020.

[8] W3C, «Data Catalog Vocabulary (DCAT) - Version 2,» [En línea]. Available: https://www.w3.org/TR/vocab-dcat-2/.

[9] W3C, «ODRL Information Model 2.2,» 2018. [En línea]. Available: https://www.w3.org/TR/odrl-model/.

[10] I. Association, «USAGE CONTROL IN THE INTERNATIONAL DATA SPACES,» International Data Space Association, Berlin, 2019.

[11] B. W. Amit Sahai, «Fuzzy Identity Based Encryption,» *IACR Cryptology ePrint Archive,* p. 86, 2004.

[12] A. Michalas, «The lord of the shares: combining attribute-based encryption and searchable encryption for flexible data sharing,» de *Proceedings of the 34th {ACM/SIGAPP} Symposium on Applied Computing, {SAC} 2019, Limassol, Cyprus, April 8-12, 2019*, https://doi.org/10.1145/3297280.3297297, 2019, pp. 146--155.

[13] The DataVaults Consortium, "D2.1 - Security, Privacy and GDPR Compliance for Personal Data management", 2020.

[14] ISO/WD TR 23644, "Blockchain and Distributed Ledger Technologies - Overview of Trust Anchors for DLT-based Identity Management". https://www.iso.org/standard/81773.html

[15] The DataVaults Consortium, "D1.3 - DataVaults MVP and Usage Scenarios", 2020.

[16] The DataVaults Consortium, "D5.2 - System Architecture, Bundles Placement Plan and APIs Design", 2021.

[17] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://doi.org/10.1007/s10838-008-9062-0.

[18] Li G., Li L., Choi T.M., Sethi S.P. Green supply chain management in Chinese firms: Innovative measures and the moderating role of quick response technology. J. Oper. Manage. 2019 doi: 10.1002/joom.1061.

[19] Sheel A., Nath V. Effect of blockchain technology adoption on supply chain adaptability, agility, alignment and performance. Manage. Res. Rev. 2019;42(12):1353–1374.

[20] Betti Q., Khoury R., Hallé S., Montreuil B. Improving hyperconnected logistics with blockchains and smart contracts. IT Prof. 2019;21(4):25–32.

[21] Yoo M., Won Y. A study on the transparent price tracing system in supply chain management based on blockchain. Sustainability. 2018;10(11):4037

[22] Christodoulou P., Christodoulou K., Andreou A. A decentralized application for logistics: Using blockchain in real-world applications. Cyprus Rev. 2018;30(2):171–183.

[23] Chang Y., Iakovou E., Shi W. Blockchain in global supply chains and cross border trade: a critical synthesis of the state-of-the-art, challenges and opportunities. Int. J. Prod. Res. 2019:1–18.

[24] Astill J., Dara R.A., Campbell M., Farber J.M., Fraser E.D., Sharif S., Yada R.Y. Transparency in food supply chains: A review of enabling technology solutions. Trends Food Sci. Technol. 2019;91:240–247.

[25] Clark B., Burstall R. Blockchain, IP and the pharma industry—how distributed ledger technologies can help secure the pharma supply chain. J. Intell. Property Law Practice. 2018;13(7):531–533.

[26] Kirkman and R. Newman, ‘‘A cloud data movement policy architecture-based on smart contracts and the ethereum blockchain,’’ inProc. IEEE Int.Conf. Cloud Eng. (IC2E), Apr. 2018, pp. 371–377.

[27] Yuanyu Zhang, Mirei Yutaka, Masahiro Sasabe and Shoji Kasahara. Attribute-Based Access Control for Smart Cities: A Smart Contract-Driven Framework. arXiv:2009.02933. 2020.

[28] QuantumMechanic “Proof of stake instead of proof of work”.[Online]Available:https://bitcointalk.org/index.php?topic=27787.0

[29] Larimer, D. (2014). Delegated proof-of-stake (dpos). *Bitshare whitepaper, Retrieved March 31, 2019, from* http://docs.bitshares.org/bitshares/dpos.html.

[30] Proof of burn. From https://en.bitcoin.it/wiki/Proof_of_burn

[31] Ren, Ling and Devadas, Srinivas "Proof of Space from Stacked Ex-panders" Proceedings, Part I, of the 14th International Conferenceon Theory of Cryptography - Volume 9985, 2016.

[32] I. Bentov, A. Mizrahi, M. Rosenfeld. Proof of activity: extending Bitcoin's Proof of Work via Proof of Staek. IACR Cryptology ePrint Archive, 452 2014.

[33] M. Castro and B. Liskov. "Practical Byzantine fault tolerance andproactive recovery". ACM Transactions on Computer Systems,20(4):398461, Nov. 2002.

[34] "Hyperledger Fabric".[Online]Available:https://www.hyperledger.org/projects/fabric A.

[35] "Hyperledger Sawtooth". [Online] Available:https://www.hyperledger.org/projects/sawtooth A.

[36] "Hyperledger Burrow". [Online] Available:https://www.hyperledger.org/projects/hyperledger-burrow.

[37] "Hyperledger Iroha".[Online]Available:https://www.hyperledger.org/projects/iroha

[38] "Hyperledger Indy".[Online]Available:https://www.hyperledger.org/projects/hyperledger-indy

[39] Popov S., "The Tangle", [Online] Available: https://assets.ctfassets.net/r1dr6vzfxhev/

2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota143.pdf

[40] LeMahieu, C."Nano: A Feeless Distributed CryptocurrencyNetwork". [Online] Available: https://nano.org/en/whitepaper.

[41] In Search of an Understandable Consensus Algorithm (Extended Version). https://raft.github.io/raft.pdf

[42] Mihir Bellare, Phillip Rogaway: Collision-Resistant Hashing: Towards Making UOWHFs Practical. CRYPTO 1997: 470-484.

[43] Merkle, R. C. (1988). "A Digital Signature Based on a Conventional Encryption Function". Advances in Cryptology — CRYPTO '87. Lecture Notes in Computer Science. 293. p. 369.

[44] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, Haixia Shi: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. J. Cryptology 21(3): 350-391 (2008).

[45] Mihir Bellare, Alexandra Boldyreva, Adam O'Neill: Deterministic and Efficiently Searchable Encryption. CRYPTO 2007: 535-552.

[46] David Cash, Stanislaw Jarecki, Charanjit S. Jutla, Hugo Krawczyk, Marcel-Catalin Rosu, Michael Steiner: Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries. CRYPTO (1) 2013: 353-373.

[47] Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2): 281-308 (1988).

[48] E. F. Brickell, J. Camenisch and L. Chen, “Direct anonymous attestation,” in ACM Conference on Computer and Communications Security (CCS), 2004.

[49] J. Whitefield, L. Chen, T. Giannetsos, S. Schneider and H. Treharne, “Privacy-Enhanced Capabilities for VANETs using Direct Anonymous Attestation,” in IEEE Vehicular Networking Conference (VNC), 2017.

[50] ETSI. Trust and Privacy Management, 2012. http://www.etsi.org/deliver/etsi_ts/ 102900_102999/102941/01.01.01_60/ts_102941v010101p.pdf [Online; accessed 26-August-2017].

[51] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. SIAM Journal on computing, 1989.

[52] Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant pay-ments. https://lightning.network (2015), accessed: 2017-05-12

[53] R. Khalil, A. Zamyatin, G. Felley, P. Moreno-Sanchez, and A. Gervais.Commit-Chains: Secure, ScalableOff-Chain Payments. Cryptology ePrint Archive, Report 2018/642.https://eprint.iacr.org/2018/642. 2018.

[54] Avarikioti, G., Kogias, E.K., Wattenhofer, R.: Brick: Asynchronous state chan-nels. arXiv preprint arXiv:1905.11360 (2019).

[55] E. Heilman, F. Baldimtsi, and S. Goldberg, “Blindly signed contracts:Anonymous on-blockchain and off-blockchain bitcoin transactions,” inFinancial Cryptography and Data Security - FC 2016 InternationalWorkshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados,February 26, 2016, Revised Selected Papers, 2016, pp. 43–60.

[56] Eberhardt and S. Tai, “On or off the blockchain? insights onoff-chaining computation and data,” inService-Oriented and CloudComputing - 6th IFIP WG 2.14 European Conference, ESOCC 2017,Oslo, Norway, September 27-29, 2017, Proceedings, 2017, pp. 3–15.

[57] Benet, J.: IPFS - content addressed, versioned, P2P file system. CoRRabs/1407.3561 (2014), http://arxiv.org/abs/1407.3561

[58] Tŕ on, V., Fischer, A., Nagy, D.A., Felḟ oldi, Z., Johnson, N.: Swap, swear and swin-dle - incentive system for swarm (2016).

[59] Bin Liu, XiaoLiang Yu, Xiwei Xu. EthDrive: A Peer-to-Peer Data Storage with Provenance,in:CEURProceedings,2017,pp.9–18.

[60] T. Mcconaghyetal. BigchainDB: A Scalable Blockchain Database (DRAFT).In:BigchainDB(2016),pp.1–65.

[61] Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable colli-sion resistance to succinct non-interactive arguments of knowledge, and backagain. In: Proceedings of the 3rd Innovations in Theoretical Computer Sci-ence Conference. pp. 326–349. ITCS ’12, ACM, New York, NY, USA (2012),http://doi.acm.org/10.1145/2090236.2090263.

[62] Parno, B., Howell, J., Gentry, C., Raykova, M.: Pinocchio: Nearly practical verifi-able computation. In: Security and Privacy (SP), 2013 IEEE Symposium on. pp.238–252. IEEE (2013).

[63] Benedikt B ̈unz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bullet-proofs: Short proofs for confidential transactions and more. In2018 IEEE Symposium on Security and Privacy,SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA.

[64] Eli Ben-Sasson, Iddo Ben-Tov, Yinon Horesh, and Michael Riabzev. Scalable, trans-parent, and post-quantum secure computational integrity.https://eprint.iacr.org/2018/046.pdf, 2018.

[65] [2017_https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data] [2020_Discovering Business Models of Data Marketplaces].

[66] [M. Spiekermann, D. Tebernum, S. Wenzel, B. Otto: A metadata model for data goods, in: P. Drews, B. Funk, P. Niemeyer, L. Xie (eds.), Multikonferenz Wirtschaftsinformatik (MKWI), 2018, pp. 326-337]

[67] [2016_A classification framework for data marketplaces]

[68] [Nieschlag, R., Dichtl, E., Hörschgen, H.: Marketing, 17th edn. Duncker & Humboldt, Berlin (1994)]

[69] 2018_https://towardsdatascience.com/data-marketplaces-the-holy-grail-of-our-information-age-403ef569fffb

[70] Stahl, F., F. Schomm, and G. Vossen, The data marketplace survey revisited, Working Papers, ERCIS - European Research Center for Information Systems, No. 18, 2014.

[71] 2018_Wibson-Technical-Paper-v1.1

[72] 2020_Discovering Business Models of Data Marketplaces

[73] **2020_Discovering Business Models of Data Marketplaces**

[74] 2017_How to Balance Privacy and Money through Pricing Mechanism in Personal Data Market

[75] 2019_A Conceptual Marketplace Model for IoT Generated personal Data

[76] 2016_Monetizing Personal Data_A Two-Sided Market Approach

[77] 2014_A Theory of Pricing Private Data

[78] 2018_Wibson: A decentralized marketplace empowering individuals to safely monetize their personal data

[79] 2016_Monetizing Personal Data_A Two-Sided Market Approach

[80] 2017_When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies.