KRAKEN

BROKERAGE AND MARKET PLATFORM FOR PERSONAL DATA

D5.4 Final KRAKEN marketplace integrated architecture document

www.krakenh2020.eu

This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 871473

D5.4 Final KRAKEN marketplace integrated architecture document

Grant agreement	871473
Work Package Leader	Lynkeus
Author(s)	Davide Zaccagnini, Minos Garofalakis, Alexandros Tragkas (LYNKEUS)
Contributors	Rob Holmes (TEX), Donato Pellegrino (TEX), Tilen Marc (XLAB), George Pikramenos (LYN), Alex Tragkas (LYN)
Reviewer(s)	Stephan Krenn (AIT), Javier Presa (ATOS)
Version	Final
Due Date	31/12/2021
Submission Date	03/02/2022
Dissemination Level	Public

Release History

Version	Date	Description	Released by
v0.1	01/03/2022	Initial version	Davide Zaccagnini
v0.2	10/01/2022	Contributions from Davide Porro (ICERT), Tilen Marc (AIT). Donato Pellegrino and Rob Holmes (TEX)	Davide Zaccagnini
V0.3	17/01/2022	Reviews by Stephan Krenn (AIT) and Javier Presa (ATOS)	Davide Zaccagnini
V0.4	25/01/2022	Version addressing reviewers comments	Davide Zaccagnini
V0.5	02/02/2022	Format changes	Davide Zaccagnini
V1.0	03/02/2022	Submitted version	ATOS

Table of Contents

List of Figures

List of Acronyms

Acronym	Description
DID	Decentralized IDentifiers
DT	Depute Tool
GDPR	General Data Protection Regulation
ID	Identifier
KCIT	KRAKEN Company Identification Tool
PPA	Privacy Preserving Analytics
SSI	Self-Sovereign Identity
SMPC	Secure Multi Party Computation
SW	Software
VC	Verifiable Credential
WP	Work Package

Executive Summary

Entering the final period of the project the KRAKEN Marketplace architecture is now finalized in all its components. The intermediate architecture was described in D5.3[1] which focused primarily on the integration of the SSI components, the design of the front-end and back-end, and the integration with the educational data infrastructure. This document focuses on the other hand on the extensions subsequently designed, namely the mobile apps, the infrastructure to assign and control institutional affiliation for individual users and the integration with the SMPC components.

Based on feedback received on previous iterations of the architecture and during the periodic review, the updated architectural scope now includes **functionalities aimed at reducing the risk of fraudulent use of the marketplace** especially in the areas of forged or otherwise tampered data products offered for sale by malicious actors. In that sense the permissioning system running on the Lynkeus blockchain was extended to include data provenance tracking functionalities also supporting the staking (escrow) against the quality of a data product.

A challenging architectural decision was resolved after intense and protracted discussions with partners regarding the implementation of two apps instead of an integrated one, encompassing both the SSI and the marketplace functionalities. While the resulting choice may slightly decrease the overall usability of the system, the separation better guarantees security of the authentication process leaving the SSI app as general-purpose identity management module.

A major focus in recent design work which led, in our view, to a highly innovative solution, is the integration of the SMPC with the marketplace front-end and back-end. The resulting architecture now defines an end-to-end secure, pay-for-computation system which, leveraging the underlying permissioning layer, orchestrates distributed computations in conjunction with a payment system powered by the Streamr DATA token. To our knowledge this is the first real world implementation of a system that directly ties the extent of data access to the economic value of the information that is gathered from it using a token-based payment system. In this regard the **KRAKEN Marketplace realizes a working implementation in which privacy, the value of information and its actual price are all technically and operationally connected.**

[1] KRAKEN Consortium: D5.3 Initial KRAKEN marketplace integrated architecture document

1 Introduction

1.1 Purpose of the document

This is a WP5 deliverable which defines the final architecture of the KRAKEN Marketplace. It's meant to serve as the description of how the marketplace has been designed in all its integrated components and the reasons for specific architectural choices.

The document builds on top of deliverable D5.3[2] describing the intermediate marketplace architecture, focusing only on the additional integrated components that were not, or not fully covered in that previous text. Besides reporting purposes, the document will serve as a reference for development activities currently under way and for the remaining part of the project to guide also activities in WP3 and WP4.

1.2 Structure of the document

The document refers as needed to D5.3 for already documented architectural designs. It is therefore structured in 3 main sections focusing respectively on:

1. The extended permissioning layer including both the data provenance tracking system and institutional affiliation credentials management (Section 2.1)
2. The SMPC and pay-for-computation systems (Section 3)
3. The marketplace mobile app architecture (Section 4)

To provide background and sufficient references to modules and integrated architectures underpinning this final version, the first part of the document presents sections of D5.3.

[2] KRAKEN Consortium: D5.3 Initial KRAKEN marketplace integrated architecture document

2 The KRAKEN Data marketplace (From D5.3)

The KRAKEN marketplace architecture consists of three main functional areas, as indicated in the diagram below. These are:

1. The permissioning layer where data access is controlled leveraging the Lynkeus Hyperledger Fabric Blockchain
2. The data access layer providing multiple infrastructures and methods allowing secure and private access to data products including the SMPC system, the TEX streaming data infrastructure and the batch data exchange system developed in the first period of the project.
3. The transaction management layer featuring technologies supporting user workflows, payments and fulfillment, mostly leveraging the TEX, Streamr marketplace.

These three layers are functionally integrated to first grant or deny data access based on the legally binding rights, then provide such access on three different modalities (SMPC, batch or streaming), and then monitoring the fulfillment of all key transaction steps. Figure 1 The KRAKEN Technology Stack From a functional perspective (see below) the Marketplace API, i.e. the back-end, connects both the desktop and the mobile apps to all other components. In particular, SSI identities are passed to the data access layer on which permissions are computed. Positive access decisions are passed through the API module to the data access layer and then to the xDai payment and fulfilment system.

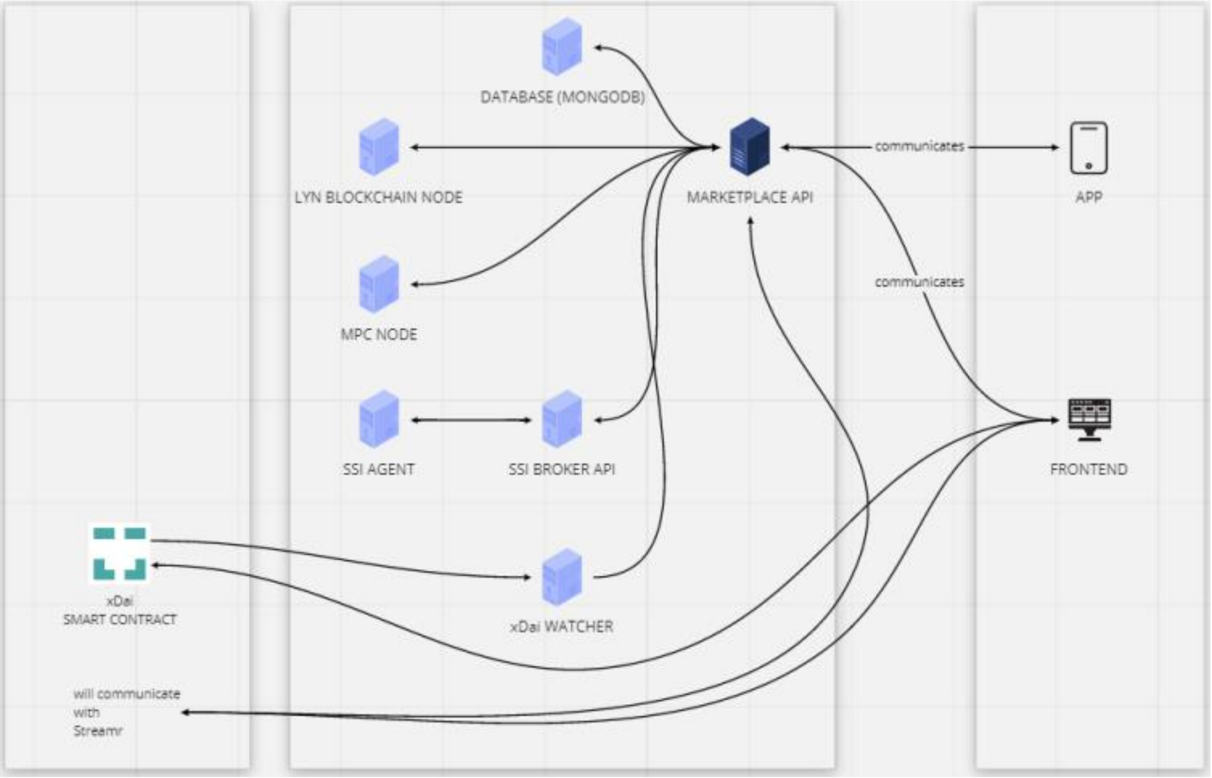

Figure 1: The marketplace architecture

This architecture is a result of parallel and iterative design efforts to link multiple modules and infrastructures, each at different stages of technological maturity. These include the Streamr marketplace, the Lynkeus data access layer, the Self-Sovereign Identity system, the Verified Credentials infrastructure and the data protection layer, which is itself composed of the SMPC system and ad-hoc data protection modules (ex. Batch data encryption). The guiding principle of such design, and indeed of the project itself, is to implement true decentralisation throughout the marketplace and the overall platform itself while at the same time providing the highest level of privacy protection for its users and the data they will exchange. Compliance with national and European privacy laws has been, in this view, a key concern in the development of this integrated system, in strict conjunction with WP7, intermediate designs, and implemented components with their integrations, were systematically reviewed from a legal and ethical standpoint following a privacy by design approach. This work is still ongoing as new modules and UI extensions are added with the final aim at automating the enforcement, by the platform itself, of legally and ethically binding terms users can enforce for the temporarily access and process of personal data for predefined purposes. From a functional standpoint the architecture is divided in three areas:

4. Permission management, mostly implemented by the Lynkeus Hyper Ledger Fabric blockchain in conjunction with the SSI system for the identification, authentication and credentialing of both individual and organisational users.
5. Data protection layer, which implements a variety of data security and privacy preserving modules and includes the Secure Multi Party Computation system employed for both distributed data analytics and encryption keys sharing mechanism, in addition to the standard data protection functionalities such as encryption at rest and transaction of batch and streaming data assets.
6. Data transaction management, mostly implemented through the Streamr marketplace technology which provides both user-facing and back-end functionalities, such as UIs, payments execution and control, secure transfers of streaming data, data product visualization and more

3 Extended data permissioning

The KRAKEN marketplace will operate in a global and highly dynamic data ecosystem in which the scope of functionalities developed in the first period, i.e. internal operations, will not suffice. Feedback received on the initial design indeed highlighted the need for outward focused designs which has been taken in close consideration by the developers, leading to an expansion of scope. The resulting work has designed new functionalities aimed at preventing fraudulent behaviour by future marketplace users, especially data sellers, and to guarantee the quality of data products. All key dimensions of data transactions were analysed, i.e. privacy, the quality of the information exchanged and its expected price. In all these areas the blockchain-based permissioning system plays a key role and substantial extensions were therefore implemented at that level. In particular the Lynkeus blockchain was extended:

a) to incorporate **data provenance parameters** to track the entire life cycle of a data product, including aggregated forms of the product derived from Data Unions or other data mergers,

b) to support the ***staking* mechanism**, by which the quality of data product is enforced by strongly disincentivizing fraudulent behaviour in the face of severe economic repercussions (See 3.2), and

c) to support the **pay-for-computation system** by linking, via permissions, the execution of distributed computations (SMPC) to the payment system.

3.1 Data provenance via blockchain

The system allows both platform administrators and users to track different versions of the same data set as it moves from its first origination on the marketplace by an individual or institution, to value-add processing such as curation or its integration with other datasets. Specially each Data Product is identified with a respective ID at the time it's first published on the ledger. At that moment, the number of tokens staked to guarantee the quality of the data is set and also captured on the blockchain as part of the transaction status parameters. As the buyer approves the data product, the transaction is updated and the blockchain records the event as a proof of data quality. Other events are captured using the same principle. For instance, as products are curated upon the base product, the product lineage is updated, tracked, and viewed over time.

The blockchain, in this way, will also store Data Unions products which combine multiple individual data sets. Each Data Union ID is made of the individual product IDs as seen on the figure below.

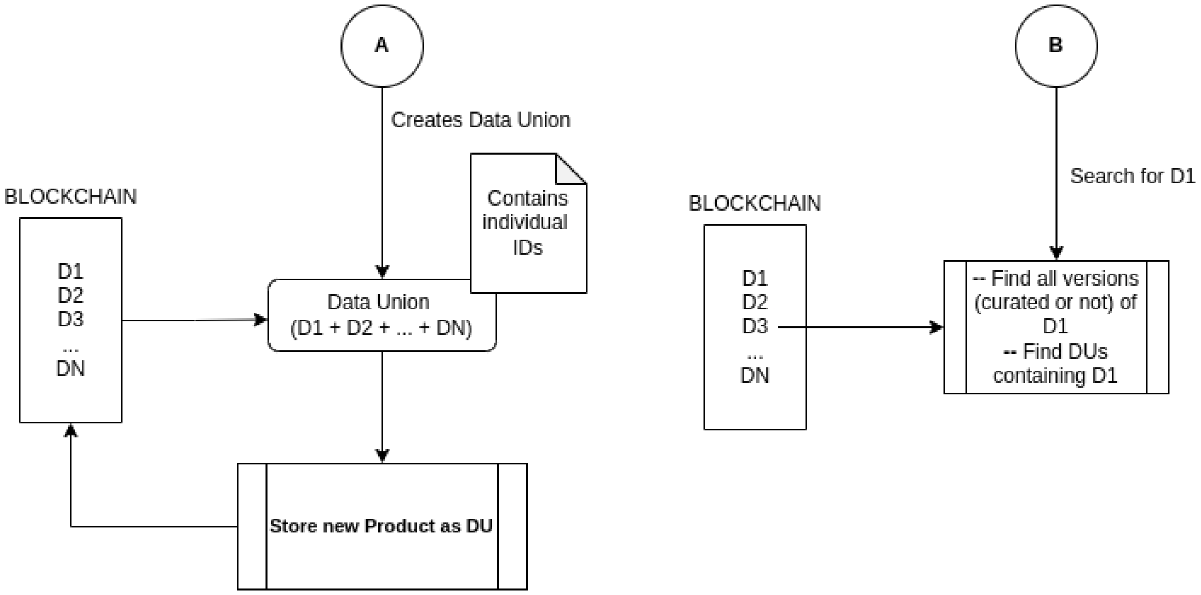

Figure 2: Data Unions and Data Products IDs

Data provenance is a foundational feature of a fully functional data quality control system and the blockchain in this sense offers an immutable, public and tamper-proof ledger which lends itself well to the purpose. Another key instrument to enforce data quality are direct incentives and deterrents to enforce data quality standards in the context of client-providers relationships. Two additional mechanisms were envisioned in this sense: the staking (escrow) systems for data products and the checking of institutional affiliations. The first act as a direct deterrent to offering poor quality or forged data on the marketplace. The second prevents the sale of institutional data outside of organizational control.

3.2 Coin staking for data quality control

Staking involves depositing in separate accounts substantial amount of cash, in the form of DATA tokens, which are released back to the seller only after the user attests to the quality of the data product. This solution addresses two key challenges: the difficulty to automate data quality control with either statistical or qualitative measures, and the related difficulty to distinguish between data sets that may share multiple characteristics and yet may be legitimately posted on the marketplace as different products by different sellers.

After extensive research by the consortium team on measures to automatically assess quality parameters in unseen datasets, the conclusion was reached that by themselves none of the existing methods offered sufficient reliability and scalability. Data utility, on one hand, is strictly dependent on the intended use of the data, i.e. the same set may be extremely valuable in one specific use case and very little in a slightly different one. Minor differences play also major roles in value assessments by data users as, for instance, the presence of a single clinical variable may dramatically increase the value of a research data set, all else being equal to similar data products.

In absence of reliable methods for assessing data quality and utility in absolute terms, the solution was identified in the staking process which will demand substantial location of DATA tokens at the time of creation of the data product. As described in D2.7[3] the tokens are relinquished back to the seller only after the buyer approves the data product for use.

Potential claims by buyers will be reviewed by platform owners and adjudicated after reviewing the data sets in question. While this mechanism requires potentially cumbersome manual review processes, by setting very high stakes for products, it acts as a powerful deterrent which will minimize

[3] KRAKEN Consortium: KRAKEN D2.7 Design for marketplace reference implementations

future claims. Limitations of this approach include the high specificity of some data sets and therefore the availability of expertise required to assess their quality which may not be readily available to platform owners. Malicious players acting as buyers may also attempt denial of service attacks through high volumes of data quality claims. These issues will be actively investigated in the remaining part of the project also in conjunction with the SSI and Crypto teams in WP3 and 4 respectively.

3.3 Institutional credential management system

In the following section the terms “institution”, “company” and “legal entity” are used as synonymous, “natural person” is used to denote a physical person, “KRAKEN platform” is used to denote the KRAKEN marketplace platform as a whole and “KRAKEN marketplace” or “KRAKEN web site” denote the pure Marketplace website SW component of the KRAKEN platform (see D2.3[4], par 4.5.1.6). The KRAKEN marketplace requires that a user, to operate on it, has been registered, moreover it doesn’t manage login from legal entities but, at the contrary, it manages only login from natural persons (for details, see “User registration process”, D2.3[5], par 3.4).

Lastly, a natural person logged to the KRAKEN marketplace can operate directly for herself or on behalf of a company, like in the case of an employee that operates on behalf of a hospital in the health pilot. In case of a natural person acting on behalf of a company, the KRAKEN platform requires that the person has previously been duly authorized by a legal representative of the company, and that the company itself has been object of an identification process by KRAKEN.

Two SW tools are developed in KRAKEN to permit the verification that a marketplace user, who claims to act on behalf of an organization, is authorized to act on behalf of said organization: the KRAKEN Company Identification Tool (KCIT) and the Depute tool.

KCIT supports the company identification process so that, a subsequent registration by a natural person within the KRAKEN marketplace, who claims to represent that organization, can be associated with the legal identity itself, this tool is deployed in a unique instance in the KRAKEN platform.

It is useful highlight here that the version of KCIT released in KRAKEN will supports the “Self-Registration” level of assurance, level evaluated sufficient for the requirements of the KRAKEN marketplace.

The user authorization process to operate on behalf of a company is supported by the Depute Tool, a single instance of the tool will be provided to every company that needs to authorize its employees to operate on behalf of the company itself.

The process of issuing of the company’s attorney to the employee is managed completely on the depute tool without using the KRAKEN web site.

Technically the Depute tool take advantage of the existing SSI infrastructure of the KRAKEN platform but in a completely transparent way to the users: also, if the tool internally uses an SSI verifiable credentials (VC) to represent the company attorney, the Attorney VCs, the company representative’s action on the Depute tool’s user interface is to authorize a company employee and not to issue a VC belonging to a specific VC schema.

After the authorization, the Attorney VC produced by the Depute tool, like every VC, will be stored inside the SSI wallet of the authorized users and it will be required by the KRAKEN marketplace when this user will access to the KRAKEN marketplace web site.

It’s important to highlight here that the usage of the SSI features in this applicative context provides a significative value to the entire process: the authorization to the user in form of VC ensures full control by the company on the authorization itself, like happens to the user’s VC login for the GDPR’s right to

[4] KRAKEN Consortium D2.3 Final KRAKEN architecture.

[5] Ibid.

be forgotten because the Attorney VC can be revoked in every moment by the company, revocation that inactivate in "real time" the authorization to the user.

3.4 Depute tool

As described above, the Depute Tool is the tool used by a company to authorize its employee on behalf of the company.

This section describes the components of the Depute Tool.

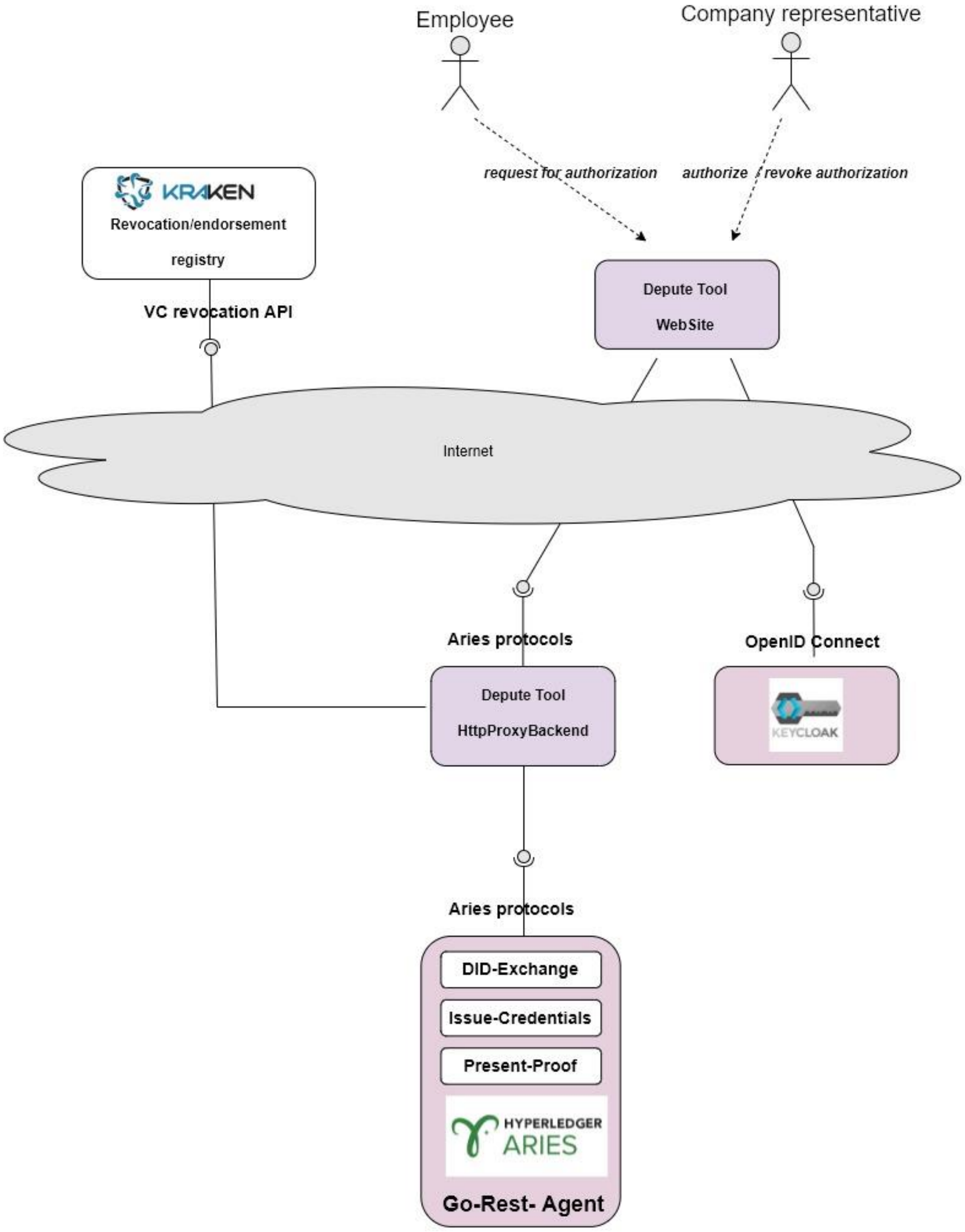

Figure 3: Depute Tool internal and external components

The Depute tool's internal components are

- the Depute Tool_WebSite, a web front end implemented in Angular 13,
- the Depute Tool _ExpressWebServer, an HTTP proxy used to protect the restful API of the go rest agent.

- the Go-Rest_agent, an open source Hyperledger Aries Go-Rest-Agent deployed without any customization
- KeyCloak, an open-source software tool implementing user authentication based on OpenID Connect.

The KRAKEN revocation and endorsement registry is external to the Depute Tool components. The Depute Tool uses the KRAKEN revocation and endorsement registry when a company representative revokes the authorization to an employee. This feature is implemented by revoking the Attorney VC that represent the authorization.

3.5 Company Identification Tool

As described above, the KCIT is the tool used by the KRAKEN platform to identify a company.

This section describes the components of the KCIT.

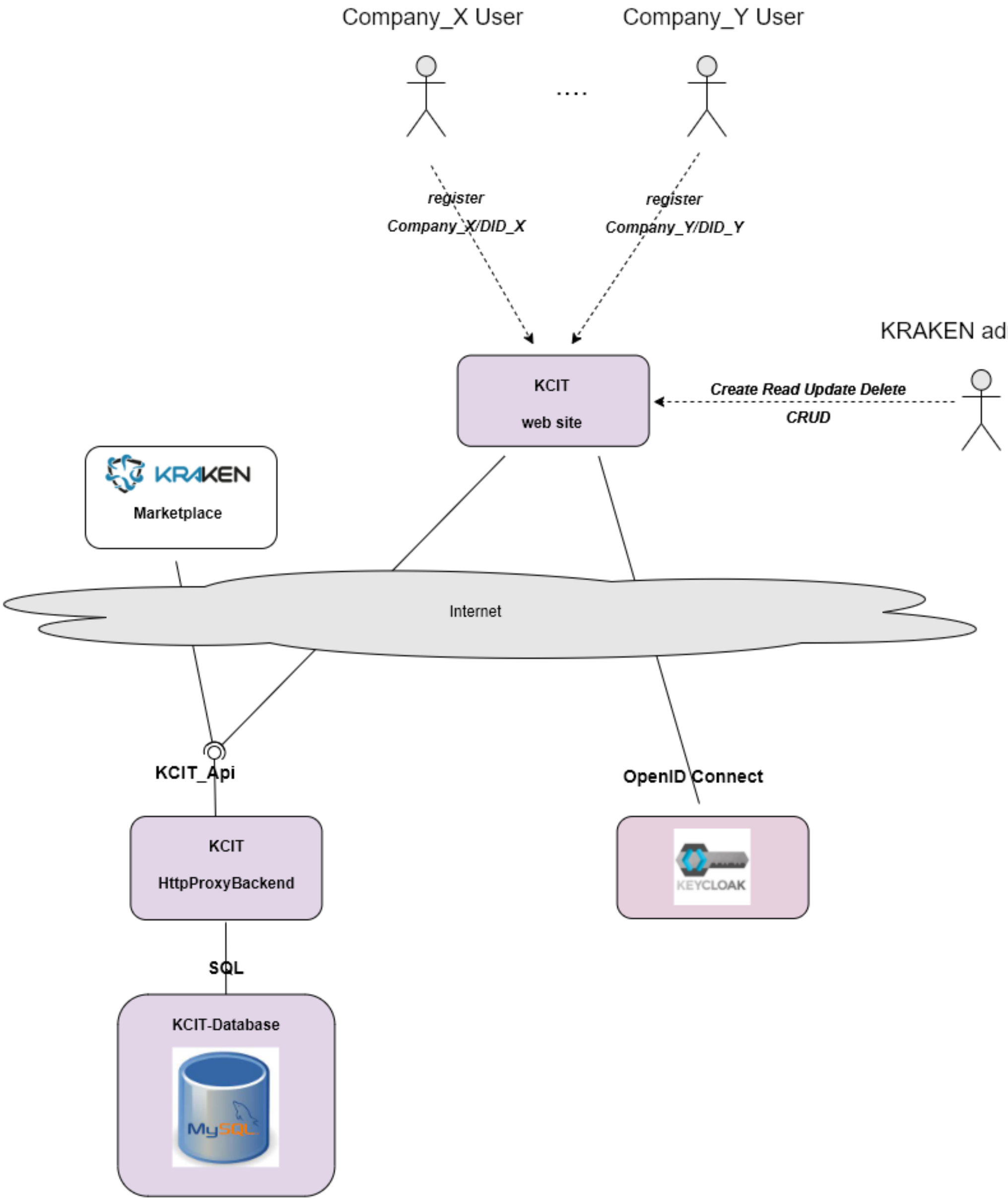

Figure 4: internal and external components

The KTIR internal components are:

- the KCIT_WebSite, a web front end implemented in Angular 13 that allows an admin to manage the configurations defined in the KCIT,
- the KCIT_ExpressWebServer, an HTTP proxy used to protect the access to the restful APIs implemented by the KCIT.
- KeyCloak, an open-source software tool implementing user authentication based on OpenID Connect.
- KCITdatabase, the container of the configuration info of the identified companies.

The only external components that uses the KCIT is the KRAKEN marketplace.

As described above, during the user registration phase on the KRAKEN marketplace, the tool permits to the KRAKEN marketplace the verification that a user, who is claiming to represent a specific organization, is providing an authorization emitted by such organization.

Also, this check takes advantage of features provided by the SSI, a consequence of the fact that the authorization is implemented using an Attorney VC issued by the company using its Depute Tool, is that the field "issuer" of the Attorney VC contains the public DID of the issuing company. The KCIT permit to KRAKEN marketplace to know which are the already identified companies and which are their public DIDs.

KCIT_Api

A CRUD (Create, Read, Update, Delete) Restful API fully accessible only by a KRAKEN admin user using the KCIT web site.

It is used also by the Company users to add her company and from the KRAKEN marketplace to list the already identified companies and to verify the issuer of an Attorney VC.

4 Integrated Secure Multi-Party Computation

4.1 Pay for computation

While apparently simple in its formulation, the pay-for-computation system is one of the main accomplishments in the KRAKEN marketplace architecture. To our knowledge this is the first real-world implementation that brings together data access permissioning, privacy preserving distributed computations, and token-based payment systems. The importance of this integration stands from its ability to answer, over time, the fundamental questions of pricing information across use cases, types of users and data life cycles. Extensive research in the problem of data valuation led our team to the conclusion that existing methods such as the Shapley Value[6], while theoretically exhaustive, were not applicable to KRAKEN or to any real-world implementation because of their impractical computational demands. For this reason, a more pragmatic approach was designed in which the balance between data value, their price and privacy metrics was going to be reached over time based on market dynamics once all the needed information is given to market players, in keeping with classic economic principles. To that end, the architecture now provides all required functionalities to efficiently arrive at that balance during short time frames after a data product is published. Specifically, the data provenance and staking mechanisms will enforce basic quality assurance and thus grounded value assessments by buyers. On top of this, the permissioning layer will provide *business intelligence* information to assess demand for certain data products, by certain types of users, for certain use cases. It is interesting to note here how these assessments sit at the intersection between privacy constraints (informed consent and intended uses, according to GDPR) and data pricing. Finally, the data provenance systems will allow sellers to study specific drivers of demand for their products in addition to price, such as added-value services that curate or aggregate the data creating more valuable offerings.

4.2 SMPC internal architecture

SMPC framework allows us to evaluate computations (functions) on data without revealing the data itself. In particular, this is achieved by splitting the data into shares, such that without knowing enough of them, no information about the data can be revealed. The shares are distributed among SMPC nodes (servers participating in SMPC network), so that they can interactively compute a function on the data without knowing the data or the result themselves. The (shares of) results are delivered to a buyer of a computation, who can merge them in the final result. It is crucial to note that with such a component KRAKEN marketplace can offer data analytics without any access to the data that is being processed and hence secures the privacy of the users by design.

The above SMPC architecture was proposed and described in D2.2[7] and D2.3[8]. The technical details about the cryptographic protocols, implementation choices and used cryptographic libraries can be found in D2.4[9] and D2.5[10], under development. Moreover, reports about the performance of the cryptographic system as well as a description of APIs to interact with it can be found in D4.3[11] Prototype implementation of cryptographic libraries. In this deliverable we focus on the integration details of SMPC with the KRAKEN marketplace.

[6] Shapley value - Wikipedia

[7] KRAKEN Consortium: D2.2 Intermediate Kraken architecture

[8] KRAKEN Consortium: D2.3 Final KRAKEN architecture

[9] KRAKEN Consortium: D2.4 KRAKEN Intermediate technical design

[10] KRAKEN Consortium: D2.5 KRAKEN final technical design

[11] KRAKEN Consortium: D4.3 Prototype implementation of cryptographic libraries

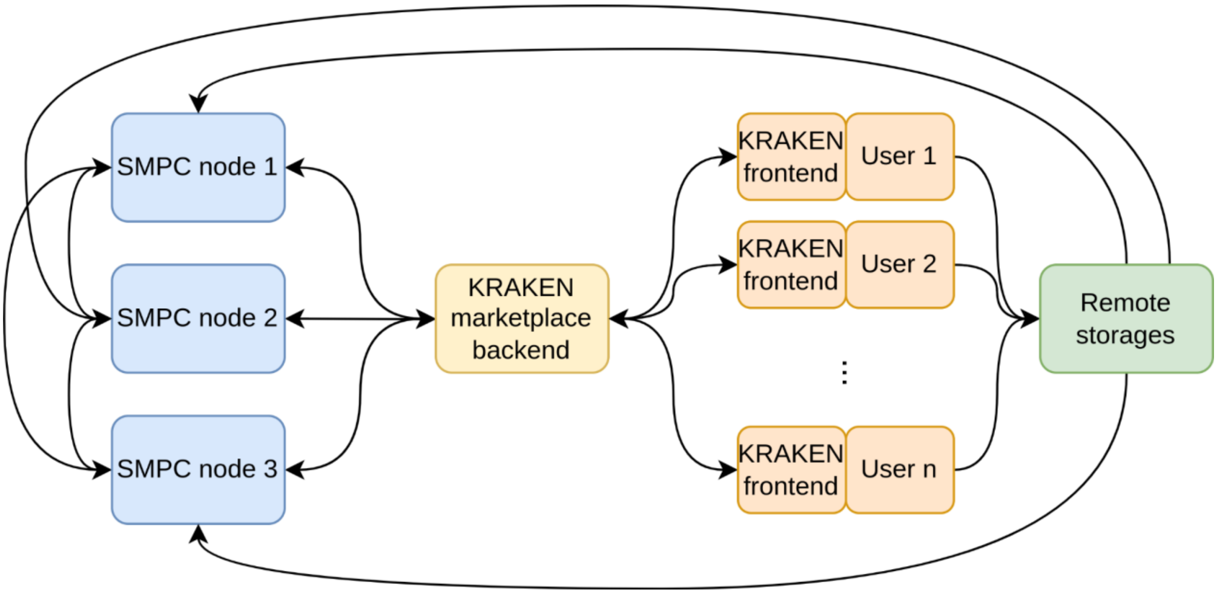

Figure 5: Internal SMPC architecture

To enable privacy preserving analytics (PPA) with SMPC, KRAKEN marketplace needs to integrate the following functionalities into its back-end and front-end:

- Frontend:
    - Publishing a data product for analytics: To offer data for PPA, a data owner needs to be able to split his/her data into shares and upload them to an external storage in an encrypted form that can be accessed only by the SMPC nodes. In particular, this task cannot be outsourced to the KRAKEN backend or any external service since it needs direct access to the dataset. Hence KRAKEN frontend provides a functionality that allows a user to load the dataset (locally) in his/her web browser, during the process of publication, and then split and encrypt the dataset using public keys of the SMPC nodes. This is implemented using WebAssembly that allows running complex programs (in our case implemented in Go) directly in a browser. Marketplace's backend receives only a link to the location of the encrypted data that it cannot access.
    - Buying a computation: Buyers presented with multiple choices of functions and datasets can request a computation on one or multiple datasets registered on KRAKEN marketplace for PPA. Since the data itself as well as the results are split in shares, the buyer needs to be able to merge the shares. Similarly as above, KRAKEN frontend provides WebAssembly based functionalities that allow the user to securely receive the shares from the SMPC nodes and merge them together in standard format such as a CSV file.
- Backend: The role of the KRAKEN marketplace backend serves solely as an intermediary between users and the decentralized SMPC nodes, with no rights to access the data or results. SMPC nodes provide an API using secured WebSockets to receive computation requests. Hence the backend needs forward users' requests to all the nodes to start a cryptographic protocol. In particular, information about the computation, links to the datasets, as well as information about the data buyer (including its public key) need to be delivered. Furthermore, the requests are recorded and checked by the SMPC nodes on a KRAKEN blockchain preventing privacy violating behaviour.

For technical details we again refer the reader to the aforementioned deliverables.

4.3 System architecture

The integrated SMPC system architecture allows users of the marketplace to perform privacy-preserving analytics on single or multiple Data Products that are listed within the marketplace data catalogue. A data provider who is concerned about the security and privacy of their data assets can create a Data Product that is only available for analytics, and receive payment in the form of the Streamr DATA token every time a data user performs a computation that involves their Data Product.

In these data transactions the marketplace acts only as an intermediary between data provider and data consumer. Content data from the Data Products are never stored by the marketplace. Instead, content data, that is made discoverable for analysis via the marketplace data catalogue, is encrypted and split into secret shares in the user's frontend environment and then stored on their cloud storage of choice.

A Data Product's secret shares can only be downloaded by the nodes in the SMPC network for computing the analytics on behalf of the data user after two important steps have been verified by the marketplace:

1) a user who wants to perform analytics has been confirmed as eligible to access the Data Product by the Lynkeus blockchain;
2) a payment notification has been received by the Marketplace from the xDai blockchain.

In the integrated SMPC system architecture which is shown in D2.7, Section 2.4.2 and repeated here for reference in Figure 6 below, the SMPC Network interfaces with the Marketplace Backend API. When a data provider using the Marketplace Frontend publishes a Data Product, the Marketplace Frontend sends the Data Product's associated metadata to the Marketplace Backend API. This includes the Data Product's descriptive information, policies and cloud storage link. The Marketplace Backend stores this information and also sends the Data Products policies to be recorded on the Lynkeus blockchain. This step is what allows the marketplace to verify that a user is able to perform analytics on the Data Product when a request for analytics is received.

On receiving requests from users to perform analytics computations on the data via the Marketplace Frontend, the Marketplace Backend API checks with the Lynkeus Blockchain that the users are eligible. If the data user is confirmed as eligible they are able to use the Marketplace Frontend to process a payment to the data provider using the Streamr DATA token on the xDai blockchain. If the payment has been successfully transferred to the corresponding Data Product owners, a notification is sent to the Marketplace Backend API by the xDai blockchain that confirms the payment.

Upon receipt of the payment notification from the xDai blockchain, the Marketplace Backend API communicates with the integrated SMPC Network to trigger the download of the encrypted secret shares from the data providers' cloud storage. As discussed earlier, this could be data from a single data provider or Data Product, or it could be data from multiple data providers or Data Products. process analytics requests. The SMPC Network finally computes the analytics and returns the results to the Marketplace Backend API. These results are encrypted specifically for the user requesting the analytics. Once the results are received by the Marketplace backend API, the user requesting the analytics then uses the Marketplace Frontend to download and decrypt the results in a CSV file format.

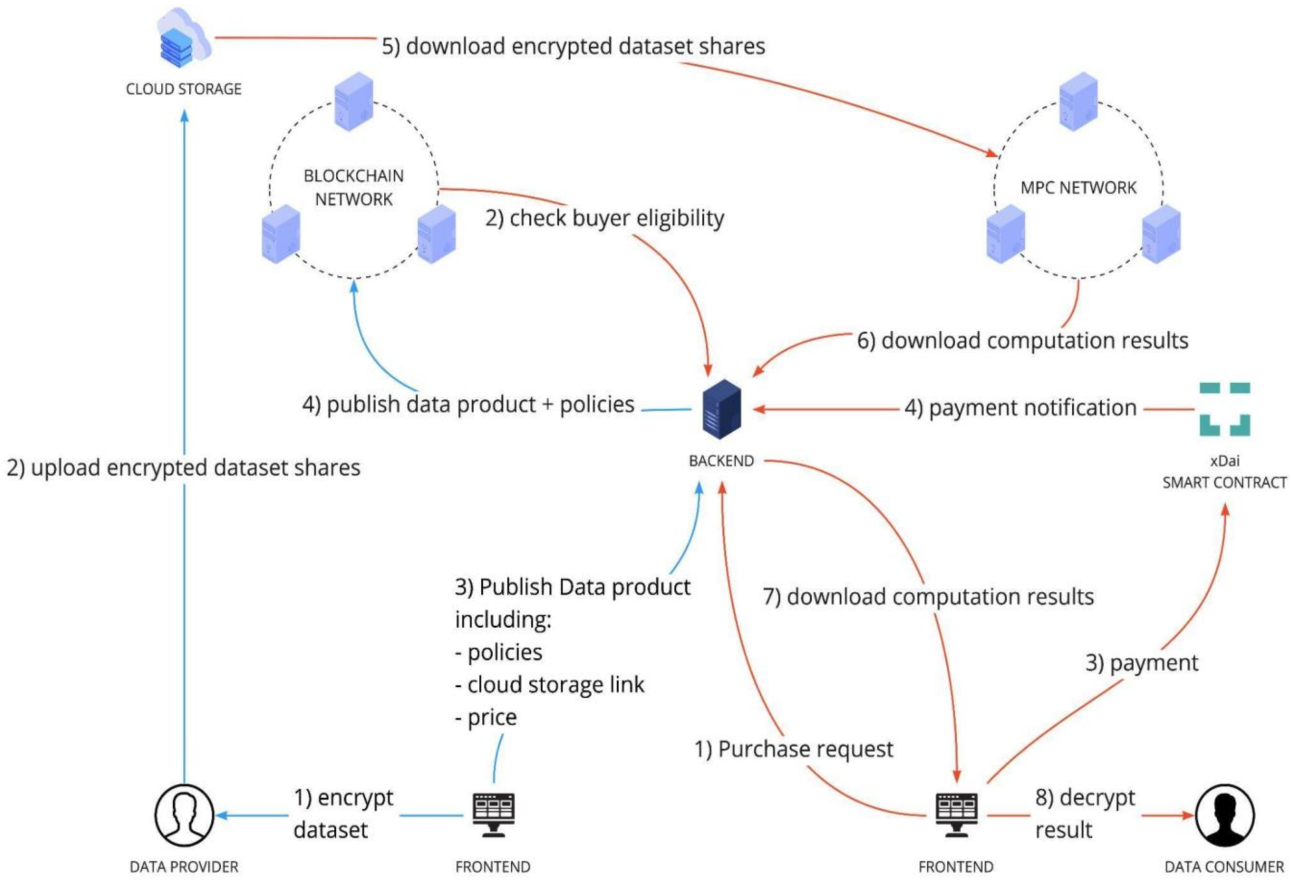

Figure 6: The marketplace's integrated SMPC system architecture. Blue: publishing process; Red: purchase and payment process

5 KRAKEN marketplace mobile application

As discussed above, after intense and protracted discussion within the consortium, a decision was made to implement two separate mobile applications serving separate and minimally overlapping workflows. In particular the SSI application has been dedicated to exclusively authenticate users while the marketplace app to actually manage data products, under the assumption that institutional users will deal with data sets on behalf of their employers mostly on the desktop version of the marketplace after logging in through the SSI app. The KRAKEN marketplace application instead will be mostly dedicated to individual users who will manage fewer and simpler personal data products, mostly from the mobile environment.

In this view both type of users will be authenticated with the SSI app, and from that moment on individual users will not need to use it again, adopting only the marketplace app on a regular basis. In this sense the design, while not optimal, creates minimal negative effects on overall systems usability and the separation better guarantees security of the authentication process leaving the SSI app as general-purpose identity management module.

The KRAKEN marketplace application connects the marketplace to its mobile environment to allow users to quickly browse data products, change permissions and availability of their own data products and see how well their data products are performing on the market. In order for the user to connect to the marketplace application the user first needs to scan a QR code to retrieve a token which will authenticate him/her. Such a QR code is made available to the user after logging in to the browser marketplace application. Once connected information is made available to the user through the backend RESTful API and http/https calls. This enables retrieving information from the marketplace. Finally, the marketplace application supports offline signing of requests to enable the editing of entries on the blockchain e.g. to change permissions or availability. That is, requests are directly signed on the application and are then sent to the backend. In this way, the expectedly frequent usage of the app, i.e. the management of data access parameters by sellers of personal data, is fully supported even directly on the app, with no additional authentication required.

6 Conclusion

The final architecture of the KRAKEN marketplace defines the integration of all necessary components to realize a fully functional system which can be deployed in real-world setting. It supports privacy-preserving tools which offer multiple data and identity protection options which users can pick from while enforcing all key tenets of the GDPR. It also supports an efficient user-friendly e-commerce experience for users in search of data assets focusing on data discoverability and ease of access. Finally, it implements an innovative infrastructure integrating blockchain-based permissioning, token-based payments and distributed computations with SMPC, which for the first time, to our knowledge, realize synergistic dynamics among privacy constrain, data value and information sharing.

KRAKEN

@KrakenH2020

Kraken H2020

www.krakenh2020.eu

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 871473