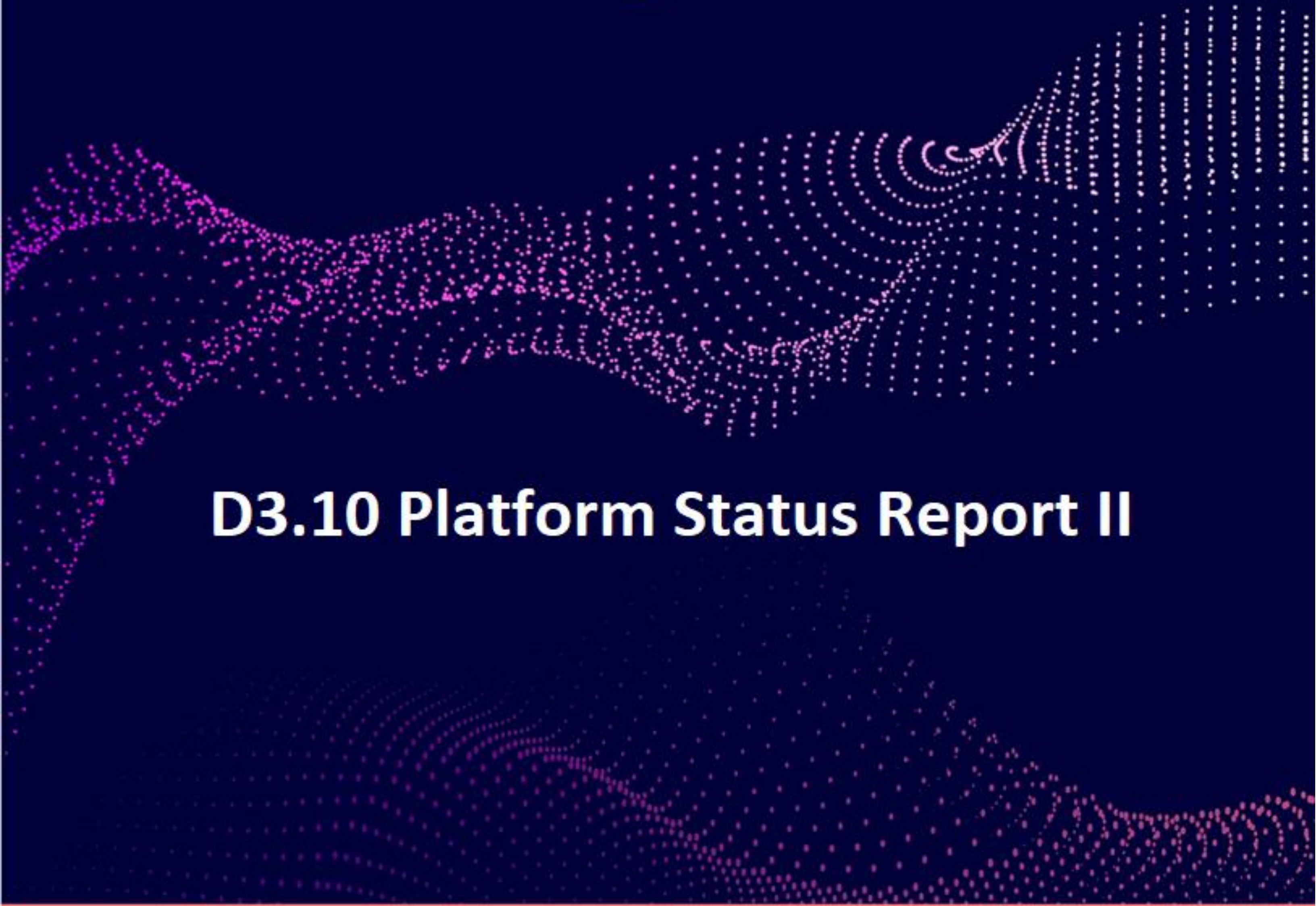
TRUSTS

Trusted Secure Data Sharing Space



# D3.10 Platform Status Report II

Authors: **Ahmad Hemid, Kim Fidomski, Benjamin Heitmann (FhG)**

December 2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 871481.

# TRUSTS Trusted Secure Data Sharing Space

# D3.10 Platform Status Report II

## Document Summary Information

| | | | |
|---|---|---|---|
| **Grant Agreement No** | 871481 | **Acronym** | TRUSTS |
| **Full Title** | TRUSTS Trusted Secure Data Sharing Space | | |
| **Start Date** | 01/01/2020 | Duration | 36 months |
| **Project URL** | https://trusts-data.eu/ | | |
| **Deliverable** | D3.10 Platform Status Report II | | |
| **Work Package** | WP3 - TRUSTS Platform implementation | | |
| **Contractual due date** | 15/12/2021 | **Actual submission date** | 15/12/2021 |
| **Nature** | Report | **Dissemination Level** | Public |
| **Lead Beneficiary** | Fraunhofer (FhG) | | |
| **Responsible Author** | Benjamin Heitmann | | |
| **Contributions from** | FhG, SWC, EMC, G1, NOVA, EBOS, LST, REL, FORTH, KNOW, RSA | | |

## Revision history (including peer reviewing & quality control)

| Version | Issue Date | % Complete | Changes | Contributor(s) |
|---|---|---|---|---|
| v0.1 | 02/10/2021 | 5 | Initial Deliverable Structure and draft ToC | Benjamin Heitmann (FhG), Kim Fidomski (FhG), Ahmad Hemid (FhG) |
| v1.0 | 15/11/2021 | 50 | Addition of entries on the current used components and the MVP.v2 platform | Victor Mireles Chavez (SWC), Nikos Fourlataras (REL), George Margetis (Forth), Ahmad Hemid (FhG) |
| v2.0 | 28/11/2021 | 70 | First draft | Ahmad Hemid (FhG) |
| v2.1 | 02/12/2021 | 85 | Additional section included | Ahmad Hemid (FhG) |
| v2.2 | 02/12/2021 | 100 | Revisions according to peer review from work package WP3 | Victor Mireles Chavez (SWC), Ahmad Hemid (FhG) |
| v2.3 | 03/12/2021 | 100 | Deliverable ready for peer review | Ahmad Hemid (FhG) |
| v2.4 | 08/12/2021 | 100 | First Peer-Review by EBOS | Gianna Avgousti (EBOS) |
| v2.5 | 08/12/2021 | 100 | Second Peer-Review by LST | Xavi Olivares (LST), Marius Parashiv (LST) |
| V2.6 | 10/12/2021 | 100 | Improvements after to Peer-Review | Ahmad Hemid (FhG) |
| V3.0 | 15/12/2021 | 100 | Final version and Submission | Benjamin Heitmann (FhG), Ahmad Hemid (FhG) |

## Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the TRUSTS consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the TRUSTS Consortium nor any of its members, their officers, employees, or agents shall be responsible or liable in negligence or otherwise however in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the TRUSTS Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

## Copyright message

## Table of Contents

## List of Figures

## List of Tables

## Glossary of terms and abbreviations used

| Abbreviation / Term | Description |
|---|---|
| ACME | Automated Certificate Management Environment |
| API | Application Programming Interface |
| CKAN | Comprehensive Knowledge Archive Network |
| DAPS | Dynamic Attribute Provisioning System |
| DevOps | Development and Operations |
| DMA | Data Market Austria |
| DSC | Dataspace Connector |
| FRs | Functional Requirements |
| GA | Grant Agreement |
| GDPR | General Data Protection Regulation |
| IDS | International Data Spaces |
| IDSA | International Data Spaces Association |
| MVP | Minimum Viable Product |
| UI | User Interface |

# Executive Summary

This report acts as a snapshot documentation of TRUSTS platform implementation status as it is continuously developed according to the DevOps best practices. This is the second of three versions of this deliverable focusing on the current state of the platform regarding its functionality and operational parameters.

Among other deliverables, this text is based on D2.7 “Architecture design and technical specifications document II” where the next iteration technical architecture of the TRUSTS platform is illustrated. This D3.10 deliverable enriches it by demonstrating the current development and implementation status of the platform as well as the core artifacts and the architecture of the next iteration platform which is implemented during the reporting period.

The development of MVP.v2 was done in accordance with the requirements previously specified, and with the initial lessons learnt, received from the WP5 after the first demonstration phase execution of the platform with the use case partners. This deliverable is based on the previous iteration version D3.9.

This deliverable includes the steps described below to specify the current platform status: the updated functional requirements (FRs) collected from the project participants who are working on tasks related to the implementation of the platform, FRs are revised by them according to their gained knowledge and expertise. The revised FRs are summarized within this document to show the changes from their previous version.

Next, the core components of this MVP version are specified. Those components form the technical architecture of MVP.v2. Due to the iterative process, the MVP.v2 architecture and the core components have slightly changed from its former architecture in D3.9. To elaborate on its workflow, the main functionalities’ workflows of MVP.v2 are presented using sequence diagrams. Meanwhile, the changes of the FRs and the platform components are pointed out in the implementation part.

The TRUSTS platform builds on the experiences and best practices from previous initiatives for supporting data markets. These are the Data Market Austria (DMA) [1] and the International Data Spaces (IDS) [2]. By reusing concepts and components developed from both DMA and IDS initiatives, realizing the strategic significance of the high-level characteristics of both will enable the TRUSTS platform to support new forms of innovation and the development of new business models.

# 1 Introduction

In the TRUSTS project, the TRUSTS platform releases are happening continuously, following the DevOps best practice principles. This deliverable (D3.10) acts like a snapshot documentation of the TRUSTS Platform. It is the second status report in a series of three reports within the lifetime of the project, due December 2021 (M24). The former was D3.9 ‘Platform Status Report I’ due for submission December 2020 (M12), resubmitted in November 2021. The latter is D3.11 ‘Platform Status Report III’ due December 2022 (M36), documenting the final version of the TRUSTS platform status report. All three versions of these reports document the existing state of the platform regarding its functionality and operational parameters.

In addition, the platform is developed in an iterative way and is based on reusing open-source software. The concept of a Minimum Viable Product (MVP) is used to allow relevant stakeholders continuously to be involved in the development of the platform.

The progress of the various tasks and work packages is monitored during the development process. Weekly sprint meetings were held in parallel with this deliverable. Technical experts and programmers intensively discussed the progress of the platform implementation, component dependencies, and obstacles. In this way, the implementation of the platform can be done in the agile development method.

The Second MVP version of the TRUSTS platform is the result of the approach mentioned above during the reported period. In addition to the selected components, the updated functional requirements (FRs), and the MVPv.2 architecture are presented and explained in this deliverable as well as its workflow. Furthermore, the development of MVPv.2 and how the changes will cover the updated FRs, and their implementation status are provided.  D3.11, the third and final version of this report, will cover MVP.v3 status.

## 1.1 Mapping Projects' Outputs

The purpose of this subsection is to map TRUSTS Grant Agreement (GA) commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

Table 1: Adherence to TRUSTS GA deliverable & tasks' descriptions

| **TRUSTS Task** | | **Respective Document Section(s)** | **Justification** |
|---|---|---|---|
| *T3.5 Platform Development & Integration* | Based on the outcomes of T2.4, this task focuses on the implementation, testing and deployment of the TRUSTS platform components. Prior to release of D2.4A, this task is expected to collaborate with T2.1, 2.2 and 2.3 in order to prepare a smooth start of development in M6. The task makes use of infrastructure provided by T3.1. Assets from existing platforms (IDS, DMA) will be reused, enhanced and adapted to cover the specifications of T2.4. This gives the task ahead by building on established and proven technologies. While, from an implementation point of view, this task covers general functionality (e.g., dataset and participant registrations), T3.2, 3.3 and 3.4 extend this functionality by providing specific state-of-the-art implementations that address the TRUSTS objectives. To that end, another goal of this work package is to integrate these contributions into a common TRUSTS platform. | Section 2<br>Section 3<br>Section 4<br>Section 5<br>Section 6 | Summary of the modified functional requirements.<br>Chosen core artifacts and components from well-known projects and initiatives<br>Architecture and workflow of MVP.v2<br>Details about the current implementation.<br>Conclusion of the deliverable, description of next actions towards the next version of this deliverable (D3.11) |
| **TRUSTS Deliverable** | | | |
| ***D3.10 Platform Status Report II***<br>*This deliverable reports on the current state of the platform regarding its functionality and operational parameters (e.g., number of integrated datasets, transactions, detected events, error rates). While platform releases are happening continuously, following the DevOps principles, these reports act as "snapshot" documentation of the platform.* | | | |

## 1.2 The Differences between this deliverable and its predecessor

Since this document is the second platform status report in a series of three reports within the lifetime of the project, the changes from its former D3.9 'Platform Status Report I' are sorted in the following table.

Table 2 compares both versions based on their content in the Sections 2-6.

Table 2: The content-based changes of this deliverable from its predecessor

| Section | D3.9 (The previous version) | D3.10 (The current version) |
|---|---|---|
| 2 | ✓ Presenting the initial functional requirements of the TRUSTS platform | ✓ Presenting the modified version of the functional requirements of the TRUSTS platform |
| 3 | ✓ Analysis of the existing artifacts from well-known initiatives and projects for the TRUSTS platform | ✓ Describing the main and core artifacts and components for the current TRUSTS platform version. |
| 4 | ✓ Reporting MVP.v0 and MVP.v1 and their architecture. | ✓ Reporting MVP.v2 architecture and its workflow. |
| 5 | ✓ Describing in detail the evolution with agile methods<br>✓ Reporting the implementation status of the components and how they cover the **initial** version FRs as well as their implementation status.<br>✓ Presenting TRUSTS Mock-ups | ✓ Describing shortly the evolution with agile methods<br>✓ Highlighting the current implementation status with changes of the components and how they cover the updated FRs as well as their implementation status.<br>✓ Snapshots of the landing page are shown. |
| 6 | ✓ Listing the project results of D3.9 deliverable and giving an overview of the next actions towards the D3.10 version. | ✓ Listing the project results of D3.10 deliverable and giving an overview of the next actions towards the D3.11 final version. |

## 1.3 Deliverable Overview and Report Structure

As a snapshot documentation of the platform's status during the reporting period, this deliverable shows the work performed in accordance with the requirements previously specified, and with the reviews received from WP5 after the execution of the first demonstration phase of the platform with use case partners. The deliverable is structured into the following sections:

2. **The updated functional requirement specifications:** The modified version of the FRs of the TRUSTS platform are summarized.
3. **Chosen core artifacts and components from well-known projects and initiatives:** The core artifacts and components that are selected for this TRUSTS platform version are described in this section. These initiatives and projects are the DMA, the IDS, and the Comprehensive Knowledge Archive Network (CKAN) project.
4. **Minimum viable products of the TRUSTS platform:** The Minimum Viable Product Version Two (MVP.v2) is described with focus on its architecture and workflow.
5. **Implementation:** The components of the TRUSTS platform and its current implementation status are listed. The changes of the platform components are prescribed and how they cover the FRs as well as their implementation status. In addition, snapshots of the landing page are shown.
6. **Conclusions and next actions:** The deliverable concludes by listing the project results which are described in this deliverable and giving an overview of the next actions towards the third version of this deliverable.

# 2 The updated functional requirements specifications

The up-to-date FRs list of the TRUSTS platform in D2.3 specify a comprehensive set of important FRs to follow. Table 3 summarizes these requirements, providing with the following headers: (1) a unique identifier number for each requirement, and (2) the description of the corresponding requirement.

Table 3: TRUSTS updated functional requirement specifications

| Req. ID | Description |
|---|---|
| **Datasets and services onboarding functionality and processes** | |
| FR1 | The system should provide standardized API descriptions for enabling providers to onboard their datasets and services |
| FR2 | The system should provide APIs that enable its interoperability/federation with other industrial marketplaces and external sources |
| FR3 | The system should be able to provide datasets and services descriptions |
| FR4 | The system should provide reference mechanisms to open data from 3rd sources, so as to make it available as an option through its data exploration, profiling and provision mechanisms. |
| **Intelligent data/service exploration and correlation functionality and processes** | |
| FR5 | The system should provide rich search mechanisms across all federated nodes for available datasets and services. |
| FR6 | The system should be able to provide datasets and services recommendations to its' users pertaining to their profile and needs |
| FR7 | The system should employ matchmaking mechanisms through which hosted datasets are matched with hosted services (e.g., suitable for their analysis or processing / customization) and vice versa. |
| Modified FR8 | The system should identify and match related datasets so as to provide combined and enriched data. |
| FR9 | The system should be able to improve datasets and services profiles based on extracted information originating from the available data |
| **Purchasing transactions and billing** | |
| FR10 | The system should provide contract mechanisms as a validation means of sellers/buyers agreements |
| Modified FR11 | The system should ensure the integrity and authenticity of the smart contracts transactions signed by its users. A dispute management process may be designed. |
| Modified FR12 | Smart contracts in the system should be accompanied by a human friendly representation (i.e natural language) |
| Modified FR13 | Mechanisms to make signed smart contracts be legally valid, enforceable and interpretable will be investigated |
| Modified FR14 | The system should encompass mechanisms for keeping transactions from being infringed |
| Modified FR15 | The system should provide the ability to connect to billing mechanisms for enabling consumers to pay providers according to the agreed smart contract |
| FR16 | The system must provide alternative and flexible pricing models taking into consideration the diversity of the available datasets and services |

| | |
|---|---|
| FR17 | The system should provide brokerage mechanisms for addressing the offerings and demands of the hosted datasets and services |
| NFR 1 | The system must provide alternative subscription contracts to the TRUSTS platform subscribers. |
| NFR 2 | The system should provide a set of templates to the asset owners to describe the T&C for using the respective asset through TRUSTS. |
| NFR 3 | Prior to procuring a dataset, information on the dataset should be provided (including data characteristics, attributes, ownership, etc.) including a sample of the data set for pre-purchased testability and preview |
| **(Meta-)Data Governance** | |
| FR18 | The system should provide explicit metadata information for each dataset or service that is accommodated in the platform |
| FR19 | The system should incorporate models, ontologies and taxonomies for the classification and semantic representation of the accommodated datasets and services in the platform |
| FR20 | The system should be able to incorporate well established or standardized ontologies from different scientific, industrial and business domains for the description of the semantic representation of the hosted datasets and services |
| FR21 | The system should provide mechanisms capable of identifying the provenance of the hosted datasets. |
| FR22 | The system should provide mechanisms capable to identify the lifecycle of the hosted datasets |
| FR23 | The system should harvest metadata from external sources. |
| FR24 | The system should be able to provide semantic information even for unstructured datasets |
| FR25 | The system should be able to keep continuously updated profiles of the hosted datasets and services based on related interactions performed with the system |
| FR26 | Dataset discovery should be based on the FAIR principle |
| **Data as a Service and Subscribers management** | |
| FR27 | TRUSTS datasets and services should be provided to the users on demand, regardless of geographic or organizational separation between provider and consumer taking into account all potential territorial legislation/regulatory restrictions. |
| FR28 | TRUSTS should be able to be deployed as a federation of distributed, interconnected and interoperable nodes. |
| FR29 | The system should enable its users to explore data and services openly, providing public descriptions. However, purchased data and services need to be exchanged point-to-point, between the seller and the buyer. |
| FR30 | The system should support mechanisms for users' (producers/consumers) subscription opting different schemes (e.g., annual, monthly, etc.) and authentication |
| FR31 | The system should support corporate accounts that fall under one subscription/enrolment per organization |
| FR32 | The system should enable authorized users to create, read, update, and delete (withdraw or make unavailable) datasets, services and user profile records |
| FR33A | The system should provide validation criteria for the new enrolled users |

| | |
|---|---|
| FR33B | The system should provide reputation/rating schemes with regard to available datasets and services. |
| FR34 | The system should allow consumers to announce their need for specific datasets / services if there are not any available, already. |
| FR35 | The system should provide notifications regarding datasets / services updates to users that have granted access to them |
| FR36 | The system should provide easy to use UIs (ensuring effectiveness, efficiency and user satisfaction) that will help users to accomplish their tasks effectively and prevent them from committing errors. |
| Modified FR37 | TRUSTS UIs and workflows must follow a business-wise rational (e.g., one stop shop), for coherently mapping the market's needs. Indicative services to be included: registration, advanced search, buy/sell data, use/provide service, browse data/service catalogue, choose contract, and upload/download datasets. |
| NFR 4 | TRUSTS UIs should be personalized |
| NFR 5 | TRUSTS should provide clear help function and documentation |
| NFR 6 | The trusts platform and service should be scalable. |
| **Data protection** | |
| Modified FR38 | The system should support collaboration between parties while preserving the privacy of the data. Methods that enables data privacy preserving on parties' collaboration will be provided by the system |
| FR39 | The system should provide de-anonymization attack assessment and risk analysis for the private / sensitive datasets to be onboard |
| Modified FR40 | The system should employ anonymization tools and guidelines for data anonymization. Information about anonymization of a dataset should be provided upon client request. |
| FR41 | The system should provide means for converting algorithms that might compromise the data privacy into safe privacy preserving ones without harming their functionality |
| **Advanced data analysis based on Machine Learning** | |
| FR42 | The system should incorporate well established ML algorithms that can be used by the TRUSTS customers for data analysis and classification. |
| FR43 | The system must incorporate a secure infrastructure for the distributed analysis of data based on ML approaches |
| **Trusted and legitimate data flows** | |
| FR44 | Mechanisms provided by the TRUSTS platform regarding personal data, non-personal data and services exploration, exchange agreements and purchase, should be compliant with the following regulations (when applicable):<br>• General Data Protection Regulation<br>• e-Privacy regulation, for electronic communications<br>• Free Flow of Non-Personal Data Regulation, for data exchange between the TRUSTS platform and subscribers<br>• Platform-to-Business Regulation, for safeguarding TRUSTS' operational transparency and fairness.<br>Mechanisms provided ensuring that local laws apply to each federated node.<br>Predefined contracts should exist. |

# 3 Chosen core artifacts and components from well-known projects and initiatives

The following five components developed in the DMA and IDS projects as well as in CKAN were reviewed in D3.9 and are chosen to be used in TRUSTS. Furthermore, the TRUST connector has been replaced with IDSA Dataspace connector based on a deep investigation and analysis where the latter has been selected because it better matches the current and future requirements of the TRUSTS platform.

For each component in the following text, a summary of the technical information of the component, regarding their dependencies, covered FRs, end-to-end functionalities, and interface descriptions as well as a short overview of the component are reported.

## 3.1. IDSA Dataspace connector

IDSA Dataspace connector is a main component for the TRUSTS platform. In the following, a short information about the IDSA Dataspace connector component (DSC) is presented. Functional requirements mentioned there are covered partially by this component for some of them. But, to have a complete FRs implementation, the platform needs other components, some of them already part of the current MPV while the remaining ones will be considered in the next iteration of the MVP(s).

It covers the main functionalities of the platform and has a rich, well documented, and easy to use REST API interface. DSC can keep all information about node's catalogs and signed agreements in the local database. Furthermore, this component has the following functionalities:

- Supports IDS Informational model
- Creation of access rules (policies) to resources (data assets)
- Creation of catalogs for data assets
- Creation of data assets
- Definition of data assets' metadata
- Creation of an offer for data assets
- Agreement's negotiation and conclusion
- Supports secured communication with another DSC and IDS metadata broker
- Sending the connector and catalog information to the metadata broker
- Searching for data assets
- Access control to local data assets
- Subscription to data asset modification
- Notification for data asset modification to the consumer
- Easy deployment using docker-compose

| **Connector** | | |
|---|---|---|
| IDSA Dataspace Connector Component | | |
| **Dependencies** | CKAN, Metadata Broker, DAPS | |
| **Functional requirements** | FR 27, FR 28, FR 29, FR 31, FR 38, FR 39, FR 40, FR 41, FR 42, FR 43, FR 44, NFR 3, NFR 6 | |
| **End-to-end functionalities** | 1 | Support metadata for asset and offer definition. Also, it can be used for a contract conclusion. |
| | 2 | API action to check access control by artifact ID and agreementID |

| | | |
|---|---|---|
| | 3 | API action to get URL by artifactID and agreementID |
| | 4 | Create catalog, resource, presentation, offer, agreement, pushing to Metadata Broker |
| | 5 | Security Present certificate for identity Mutual TLS |
| | 6 | Contract (Agreement) communication with CKAN |
| **Interface description** | ● REST API | |

## 3.2 Metadata Broker and Metadata Storage

The Metadata broker acts as a centralized repository of the metadata about the assets being exchanged on TRUSTS. It is accompanied by a metadata storage system in the form of a triplestore. Current implementation is Apache Jena Fuseki[1] for the triplestore, and the broker-open-core[2] for the metadata broker.

| **Metadata Broker** | | |
|---|---|---|
| Metadata Broker Component | | |
| **Dependencies** | - | |
| **Functional requirements** | FR 1, FR 3, FR 5, FR 18, FR 21, FR 22, FR 23, FR 25, FR 26 | |
| **End-to-end functionalities** | 1 | When a user searches for assets, this component will answer the queries |
| | 2 | When a participating organization onboards an asset, this component will concentrate its metadata so that other organizations can also discover it |
| **Interface description** | ● Exchange of messages according to the IDSCPv2 | |

## 3.3 Platform Interfaces (CKAN)

The platform interface is the main point of interactions between casual human users and the TRUSTS platform. Users who are not members of a registered organization can access this interface to perform a limited set of operations, without the need to sign up to TRUSTS and set up their own corporate node.

| **Platform Interface** | | |
|---|---|---|
| CKAN Component | | |
| **Dependencies** | Dataspace Connector, Metadata Broker | |
| **Functional requirements** | FR 1, FR 3, FR 4, FR 5, FR 25, FR 27, FR 29, FR 30, FR 31, FR 32, FR 33A, FR 33B, FR 34, FR 36, FR 37, FR 44, NFR 1, NFR 2, NFR 3, NFR 4, NFR 5 | |
| **End-to-end functionalities** | 1 | Onboarding datasets and services from the user portal node is done through the corporate interface |
| | 2 | Searching for datasets is done through the corporate interface |
| **Interface description** | UI for users:<br>● REST API used to provide access to assets it hosts<br>● Makes extensive use of the DSC REST API | |

[1] https://jena.apache.org/documentation/fuseki2/

[2] https://github.com/International-Data-Spaces-Association/metadata-broker-open-core

## 3.4 Corporate Interface (CKAN)

The corporate interface is the main point of interactions between corporate users and the TRUSTS platform. Users who are members of an organization participating in TRUSTS, can access the corporate interface deployed in their node to manage the assets they are offering to the platform, as well as those that they have been acquired from the platform.

| **Corporate Interface** | | |
|---|---|---|
| CKAN Component | | |
| **Dependencies** | Dataspace Connector, Metadata Broker | |
| **Functional requirements** | FR 1, FR 3, FR 4, FR 5, FR 27, FR 29, FR 30, FR 31, FR 32, FR 34, FR 36, FR 37, FR 44, NFR 1, NFR 2, NFR 3, NFR 4, NFR 5 | |
| **End-to-end functionalities** | 1 | Onboarding datasets and services from the user portal node is done through the corporate interface |
| | 2 | Searching for datasets is done through the corporate interface |
| **Interface description** | UI for users:<br>● REST API used to provide access to assets it hosts<br>● Makes extensive use of the DSC REST API | |

## 3.5 Landing page

The landing page is the first page that users interact with when they visit the TRUSTS platform. Users are greeted on this page with the platform's title as well as an explanation of the platform's main advantages over competitors. Snapshots for this component will be covered in the subsection 5.4 to elaborate more about its functionalities.

| **Landing Page** | | |
|---|---|---|
| Landing Page Component | | |
| **Dependencies** | Platform Interface (CKAN) | |
| **Functional requirements** | NFR 5 | |
| **End-to-end functionalities** | 1 | Individual / organization enrolment and login |
| | 2 | Password reset |
| | 3 | View partners, features and a description of the platform |
| **Interface description** | UI for non-registered users to:<br>● learn about TRUSTS platform, legal terms, contact points, etc.<br>● enroll and login<br>● reset password | |

# 4 Minimum viable products of the TRUSTS platform

As it has been described in D3.9, the Minimum Viable Product (MVP) is a functional iteration of a product with just enough features to be useful to meet a minimum number of requirements which makes it possible to quickly obtain user's feedback and further evolve. Therefore, undesirable development can be detected and prevented as early as possible. Until the writing of this text, three MVPs have been started, the first one is MVP.v0, then followed by MVP.v1, and the next and current version is MVP.v2.

The following text discusses MVP.v2 and how it differs from the previous version MVP.v1, as well as its components, architecture, and workflow.

## 4.1 MVP version 2 overview

MVP version 2, as it is the next iteration of the MVPv.1, it is worth at the beginning mentioning its main changes, compared to its previous version. The main changes are summarized, as follows:

- Replacing of Trusted Connector with Dataspace Connector
- Implementation of IDS Informational model
- Creating catalogs for data assets
- Replacing old data asset definition with new more reach definition of data assets (datasets, services, applications)
- Flexible definition of access control to data assets
- Creation an offer for data assets
- Pushing node's catalog information to central broker to make available for searching from any node
- UI interfaces to search data assets
- Contract (Agreement) conclusion between Seller and Buyer of data assets using simple UI interfaces
- Restriction of access control to data assets

As it has been specified, the MVP v2 is completely different from the MVP v1. After additional research and a big progress of the IDSA team working group on Dataspace Connector and other components, the Trusted Connector component has been replaced with the DSC component as more suitable to the FRs of the project.

In Figure 1, the structure of MVP.v2 is demonstrated as well as the links between its components. The platform is built as a network between independent nodes connected all together using secured channels of communication based on TLS mutual authentication.

The platform has three types of nodes:

i. **Corporate Node** Is used by partners of TRUSTS that have complex infrastructure. They can have many users and many services and applications that are provided or consumed via the TRUSTS platform. They can set any authorization system and communication structure inside their nodes. The TRUSTS Operator is not responsible for setting up or supporting instances of corporate nodes but will provide detailed instruction manuals and all necessary software. Among the components being set up, the Corporate Interface is a web application through which the organization's users can interact with the TRUSTS platform.

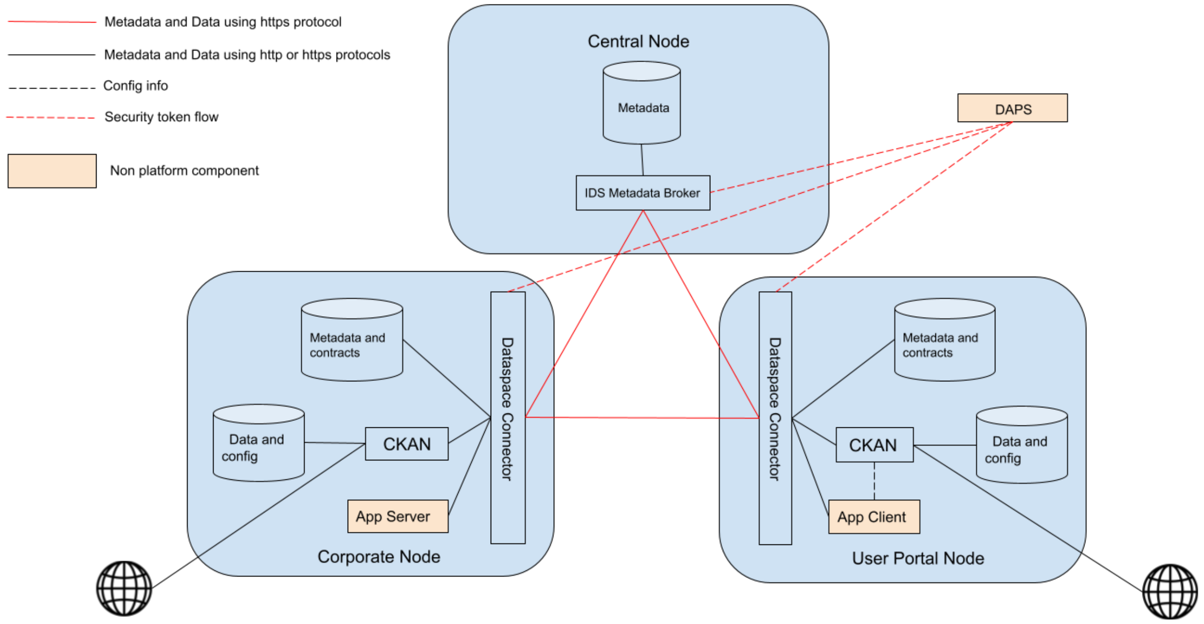


Figure 1: TRUSTS MVP.v2 Architecture

i. **User Portal Node** was created to cover the needs of individual users of the TRUSTS platform. It is set up and maintained by the TRUSTS Operator. By accessing this node, individual users can benefit from some of the functionalities of the TRUSTS platform, without the complexity of setting up a corporate node. Additionally, it is the entry point for all organizations intending to join the TRUSTS platform, as it provides landing pages, legal information, and setup instructions. One of the components included in this node is the platform Interface, which acts as the main point of access to the above-mentioned functionalities. In principle, the TRUSTS platform could have more than one User Portal Node.

ii. **Central Node** exists to support the operation of the whole TRUSTS platform, playing the role of authorization, monitoring, smart contract executor, catalog, application repository, among others. This node is created and maintained by the TRUSTS Operator.

The IDS DAPS component issues access tokens that are required to access the services and the data of other connectors. The protocol implemented in DAPS allows certified connectors to authenticate to the DAPS. In return, the connectors retrieve an access token that they can use to access other connectors. The decision about permissible access is not made by the DAPS, but by the connector that the requesting connector wants to access.

The connections between different nodes are all done using the IDSCPv2 communication protocol and are handled by the set of standardized components. Connections coming from outside the node are first received by the DSC Connector component and then distributed to other components. The IDSCPv2 protocol, apart from the standard asymmetric-key encryption of modern HTTPS connections, allows for conveying information specific for the functionalities of the platform. This includes user identification tokens, names and ports of the different nodes involved, as well as IDs of the components they are destined to.

## 4.3 MVP version 2 workflow

To show the main internal and external components' communications in the current MVP.v2 version, the following will cover the main five workflow functionalities. Each functionality has a title to explain its actual function, an explanation of the workflow, and a sequences diagram, as follows:

- **Data Asset Onboarding**
    1. Data asset onboarding process starts from creation by User in Provider node definition of data asset using CKAN UI.
    2. CKAN in response to this user's action creates all metadata for data asset in appropriate catalog and then create an offer for this data asset inside into Provider Dataspace Connector
    3. User presses button "Publish" to promote the offer of data asset to the Metadata Broker
    4. CKAN initiates the process of publishing an offer to Broker sending a push command to Provider DSC.
    5. Provider DSC requests security token from DAPS.
    6. DAPS responses with token or error.
    7. Provider DSC publishes information about itself and an offer to Metadata Broker, presenting the token.
    8. Metadata Broker checks Provider Token. If checking is OK, Broker creates records about Provider's Connector and its offers and sends back confirmation. Otherwise, Metadata broker sends error message back to Provider DSC
    9. Provider DSC forwards the Metadata Broker response to CKAN
    10. CKAN shows the result of Metadata Broker to the User's screen.

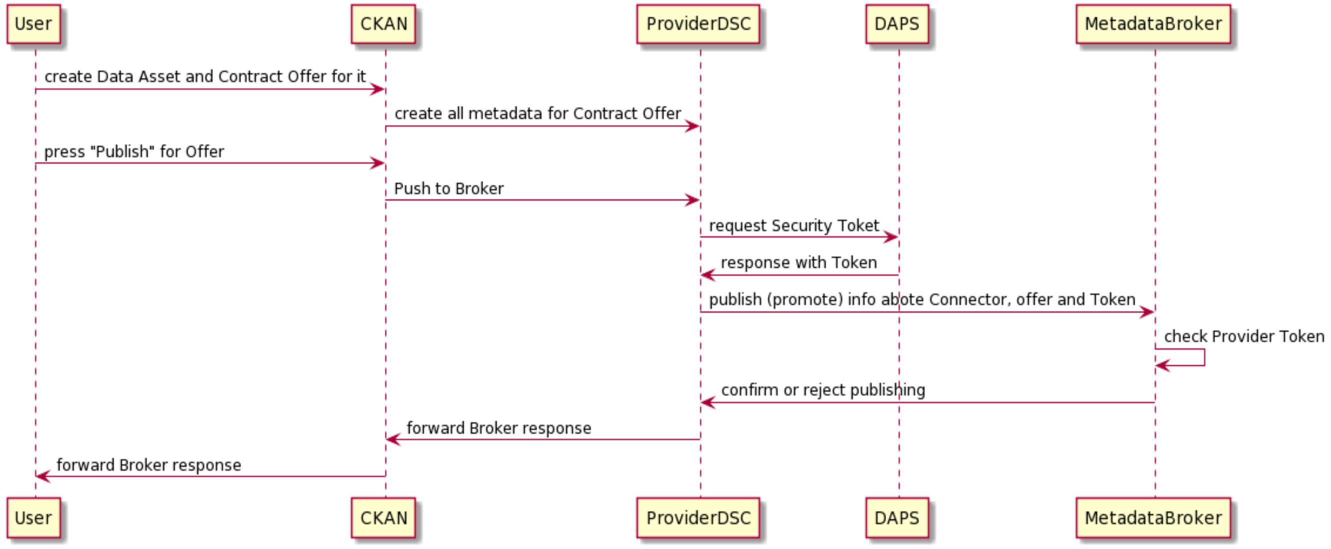


Figure 2: Data Asset Onboarding

- **Data Asset Searching**

  Precondition to start a search data asset is an existing offer for this data asset in Metadata Broker.

  1. User in Consumer node input search request for data asset in CKAN UI.
  2. CKAN forward the searching request to Consumer Dataspace Connector.
  3. Consumer DSC requests security token from DAPS.
  4. DAPS sends back a token or error if it doesn't authenticate DSC.
  5. Consumer DSC transforms search request to SPARQL query, adds its token and sends request to Metadata Broker.
  6. Metadata Broker checks Consumer Token and if it's correct executes SPARQL query.
  7. Metadata Broker returns the searching result or error to Consumer DSC.
  8. Consumer DSC forward Broker's response to CKAN.
  9. CKAN shows the search result on User's screen.

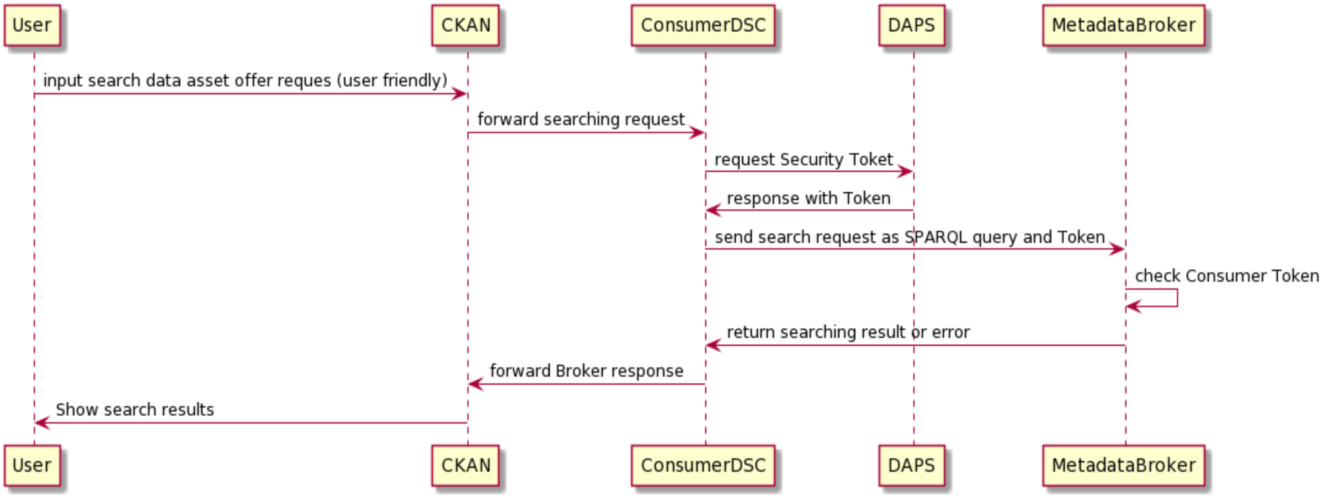


Figure 3: Data Asset Searching

- **Data Asset Agreement Conclusion**

Data asset agreement conclusion can be fulfilled after searching in Metadata Broker.

1. The User in Consumer node should select from the searching result suitable asset offer and press “Select offer”.
2. CKAN forwards the selected offer to Consumer Dataspace Connector.
3. Consumer DSC extracts information about location of the data asset.
4. Consumer DSC requests a token from DAPS.
5. DAPS sends token or error back to Consumer DSC
6. Consumer DSC sends a request for Agreement and Token to Provider DSC.
7. Provider DSC checks Provider token, if it is correct then stores information about agreement
8. Provider DSC returns finalized Agreement or error to Consumer DSC.
9. Consumer DSC store agreement information.
10. Consumer DSC forwards agreement details to CKAN.
11. CKAN shows agreement details in the User’s screen.

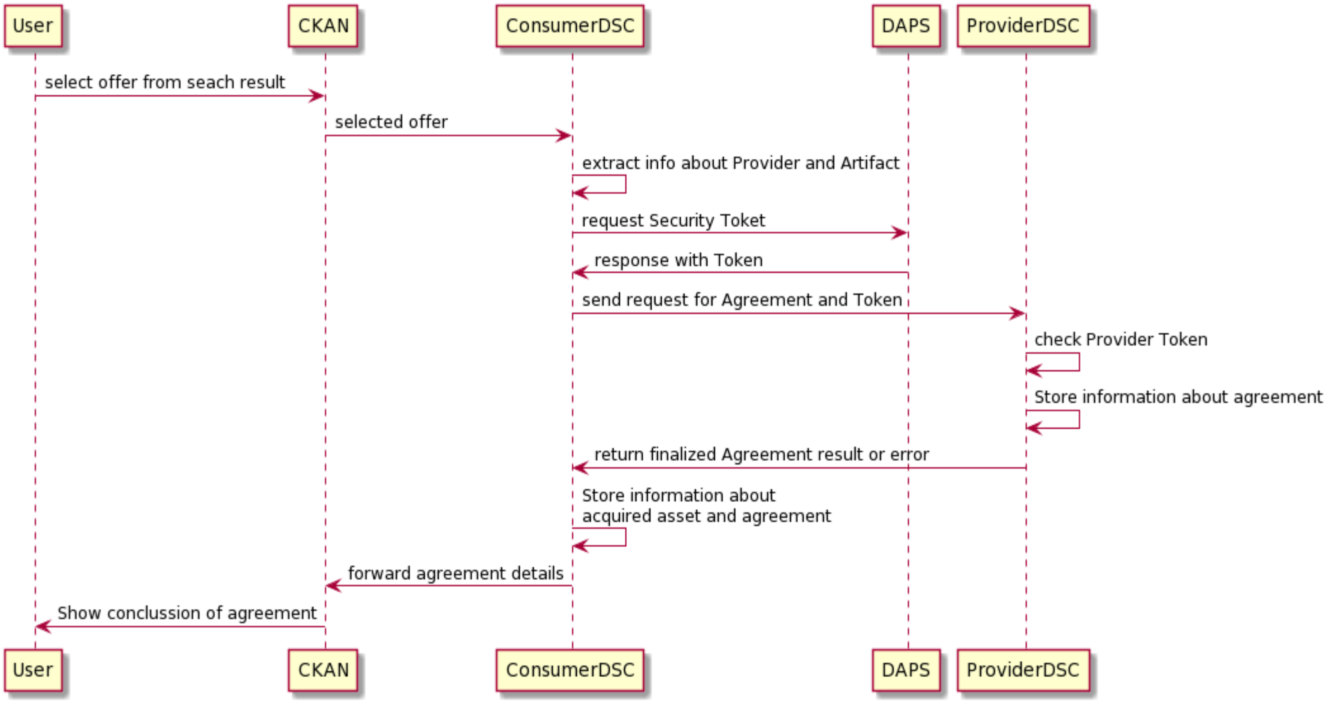

User
CKAN
ConsumerDSC
DAPS
ProviderDSC
select offer from seach result
selected offer
extract info about Provider and Artifact
request Security Toket
response with Token
send request for Agreement and Token
check Provider Token
Store information about agreement
return finalized Agreement result or error
Store information about
acquired asset and agreement
forward agreement details
Show conclussion of agreement
User
CKAN
ConsumerDSC
DAPS
ProviderDSC

Figure 4: Data Asset Agreement Conclusion

- **Data Asset (Dataset) Access**

Users of a Consumer node can download dataset from the provider only if previously the Consumer has created an agreement for access to this dataset with the provider.

1. The User of the Consumer node selects from the list of agreements (contracted data assets) the dataset for downloading and press download button.
2. Consumer CKAN forwards information about selected agreement (dataset) to Consumer DSC.
3. Consumer DSC extracts info about Provider for dataset to route request.
4. Consumer DSC request a token from DAPS
5. DAPS responses with token or error.
6. Consumer DSC sends a request for the dataset accompanied with its token to the provider.
7. Provider DSC checks the consumer token.
8. Provider DSC checks access rights for the consumer to the resource.
9. If steps above, 6 and 7 succeed, then the Provider DSC requests the dataset from Provider CKAN.
10. Provider CKAN returns the dataset to Provider DSC.
11. Provider DSC returns the dataset to the Consumer CKAN. The consumer CKAN downloads the dataset and shows it in the User browser.

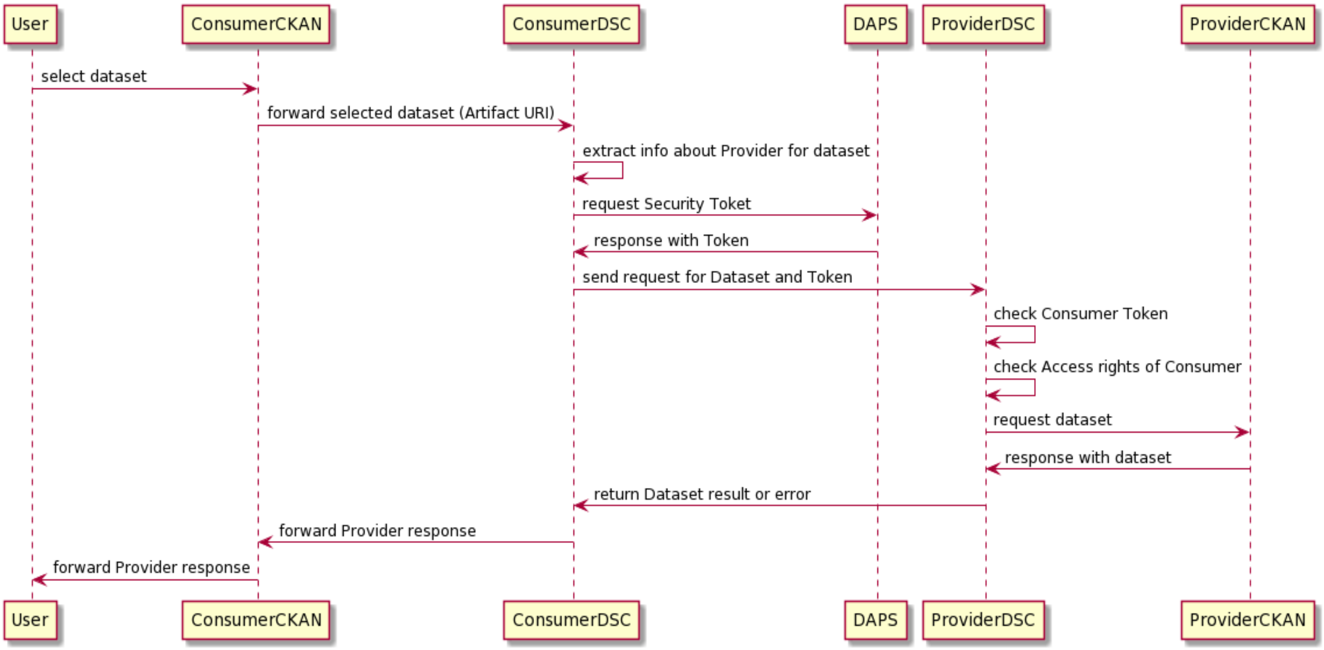

User
ConsumerCKAN
ConsumerDSC
DAPS
ProviderDSC
ProviderCKAN
select dataset
forward selected dataset (Artifact URI)
extract info about Provider for dataset
request Security Toket
response with Token
send request for Dataset and Token
check Consumer Token
check Access rights of Consumer
request dataset
response with dataset
return Dataset result or error
forward Provider response
forward Provider response
User
ConsumerCKAN
ConsumerDSC
DAPS
ProviderDSC
ProviderCKAN

Figure 5: Data Asset (Dataset) Access

- **Data Asset (Service) Access**

Users of a Consumer node can set up an application client to the provider resources only if previously the Consumer has created an agreement for access to these resources with the provider.

1. The User of the Consumer node selects from the list of agreements (contracted data assets) the services to access, then the user downloads from ConsumerCKAN all needed information for setting up an app client to access provider services.
2. Consumer CKAN responses with setting info for the app client to the user.
3. The User configures the app client and runs it.
4. The app client issues an action request for the application server through the Consumer DSC.
5. Consumer DSC extracts info about the Provider of the services.
6. Consumer DSC requests a token from DAPS.
7. DAPS responses with token or error.
8. Consumer DSC sends a request to the service accompanied with its token to the provider.
9. Provider DSC checks the consumer token.
10. Provider DSC checks access rights for the consumer to the resource.
11. If steps above, 6 and 7 succeed, then the Provider DSC forwards the request to the AppServer.
12. AppServer response with data to Provider DSC.
13. Provider DSC returns data result or error to Consumer DSC.
14. Consumer DSC forwards the provider response to the AppClient.

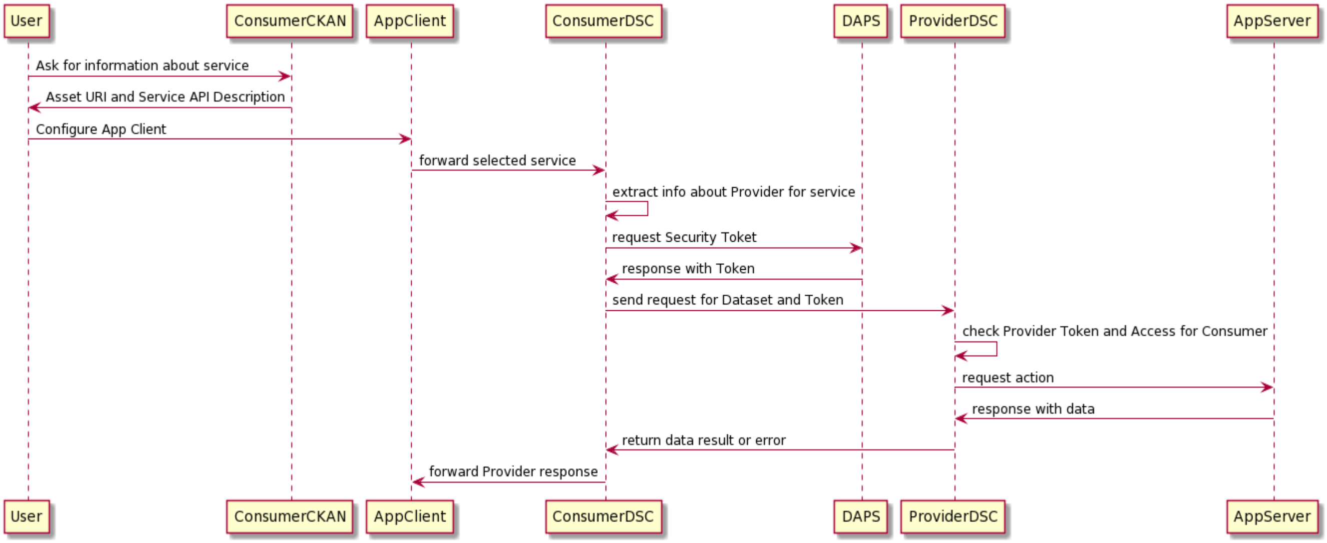

User
ConsumerCKAN
AppClient
ConsumerDSC
DAPS
ProviderDSC
AppServer
Ask for information about service
Asset URI and Service API Description
Configure App Client
forward selected service
extract info about Provider for service
request Security Toket
response with Token
send request for Dataset and Token
check Provider Token and Access for Consumer
request action
response with data
return data result or error
forward Provider response
User
ConsumerCKAN
AppClient
ConsumerDSC
DAPS
ProviderDSC
AppServer

Figure 6: Data Asset (Service) Access

- **Data Asset (Application) Access**

Users of a Consumer node can set up and run an application in a Consumer node only if previously the Consumer has created an agreement for access to these resources with the provider.

1. The User of the Consumer node selects from the list of agreements (contracted data assets) the application to access, then the user downloads from ConsumerCKAN all needed information for setting up an app client to access provider services.
2. Consumer CKAN forwards a request for the selected application to Consumer DSC.
3. Consumer DSC extracts info about the Provider of the services.
4. Consumer DSC requests a token from DAPS.
5. DAPS responses with token or error.
6. Consumer DSC sends a request for the application accompanied with its token to the provider.
7. Provider DSC checks the consumer token.
8. Provider DSC checks access rights for the consumer to the resource.
9. If 6 and 7 succeed, then Provider DSC responses with application setup info to Consumer DSC.
10. Consumer DSC forwards Provider response to Consumer CKAN.
11. Consumer CKAN shows User application setup info to User.
12. User sends a request to download the application from TRUSTS Docker images repo.
13. Docker Images Repo sends the docker image of the application to User.
14. The User set up the application.
15. The User runs the application in Consumer Node.

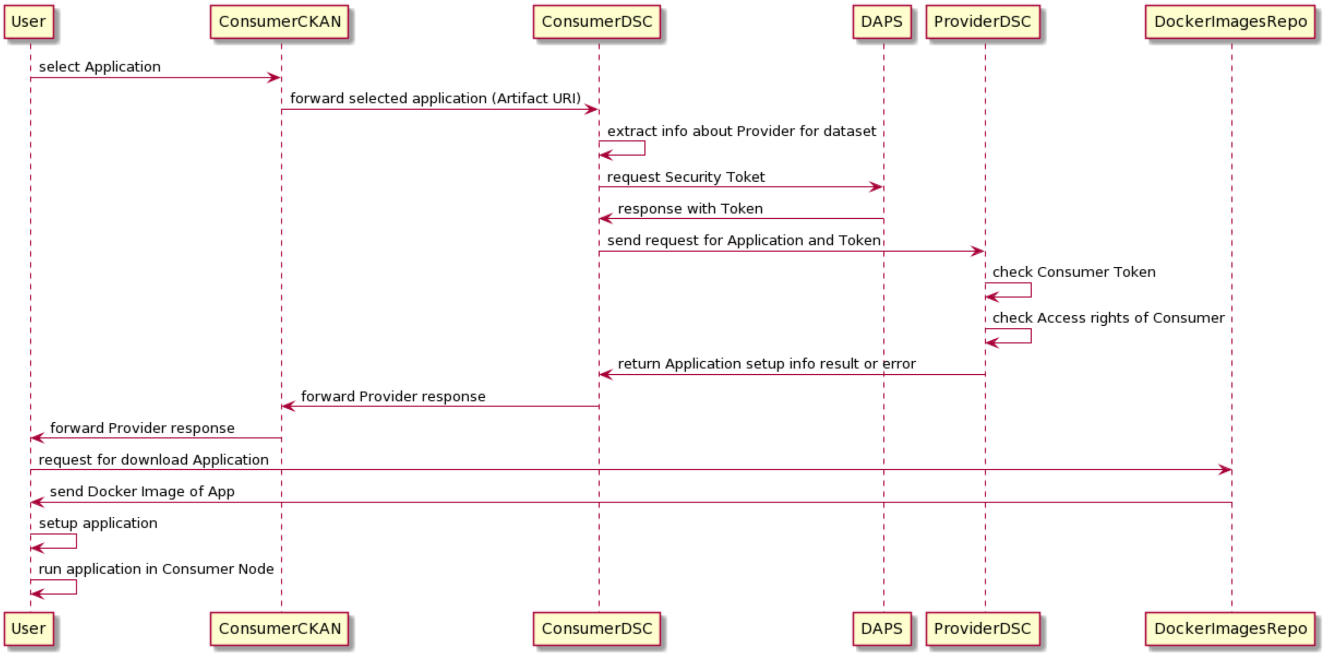

User
ConsumerCKAN
ConsumerDSC
DAPS
ProviderDSC
DockerImagesRepo
select Application
forward selected application (Artifact URI)
extract info about Provider for dataset
request Security Toket
response with Token
send request for Application and Token
check Consumer Token
check Access rights of Consumer
return Application setup info result or error
forward Provider response
forward Provider response
request for download Application
send Docker Image of App
setup application
run application in Consumer Node
User
ConsumerCKAN
ConsumerDSC
DAPS
ProviderDSC
DockerImagesRepo

Figure 7: Data Asset (Application) Access

# 5 Implementation

The following discusses the implementation process of the TRUSTS platform. First, the planning for the evolution of the platform by employing an idea from agile development is offered. Then, the recent changes of the components of the TRUSTS platform and how they are going to address the updated version of the functional requirements are demonstrated. Followed by the implementation status of those components. Finally, it is concluded by offering snapshots for the landing page component

## 5.1 Planning for the evolution of the platform with agile methods

As it has been reported in D3.9, the agile methods are used in this project achieving continuous improvement. The concept of a "Minimum Viable Product (MVP)" is used to involve relevant stakeholders in the evolution of the platform, at different points in time during the project. A MVP [3], [4] ] can be thought of as a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.

In TRUSTS, each MVP version is a clearly specified hand-off point, at which a specific combination of platform components in specific versions is released to all partners who are working on the implementation of use cases and on the implementation of privacy enhancing technologies. These partners are the internal customers of the MVP within the TRUSTS project. Every MVP version is released internally within the project, together with a description of how to perform technical tests. The results of these technical tests will then be used as input for the next version of the MVP. The goals for the MVPs are characterized by three dimensions: integration of APIs, functional integration, and operational integration, as it has been detailed in D3.9.

## 5.2 The changes of the platform components addressing the updated FRs

The following lists the changes of the planned components of the TRUSTS platform. The components are described in the second iteration technical architecture of the platform (D2.7 “Architecture design and technical specifications document II”). Some of the components have been eliminated or changed from the previously selected components in the preceding version of this document D3.9 “Platform Status Report I”, coloured with red and stroked-through.

In summary, the following components presents the current changes of the TRUSTS technical architecture:

- ~~C1 - Trusted Connector~~ C1 - Dataspace Connector
- C2 - Dataflow Router
- C3 - Reverse Proxy
- C4 - Recommender
- C5 - Platform Interface (CKAN)
- C6 - Landing Page
- ~~C7 - Asset Consumer~~
- C8 - Notification Service
- C9 - Metadata Mapper
- ~~C10 - Usage Control~~
- C11 - Mapping Builder
- C12 - Corporate Interface (CKAN)
- ~~C13 - Services Consumer Adapter~~
- C14 - Data Exchange TRUSTS Component
- C15 - Data Exchange Client Component
- C16 - Registry of Data markets
- C17 - Business Support Services
- C18 - Broker + Metadata Storage
- C19 - App Store
- ~~C20 - Identity Provider + Key Distribution System~~
- C21 - Vocabulary Services
- ~~C22 - Dynamic Attribute Provisioning System (DAPS)~~ C22 - Distributed Authorisation Component
- C23 - Automated Certificate Management Environment (ACME)
- C24 - Smart Contract Execution

In the following table, a summary of the changes of the FRs and components which address the updated version of the FRS is listed. To highlight the change in colours, for both FRs or complements: new items are coloured with green, modified ones have the blue colour, and the eliminated ones are coloured with red and stroked-through, as shown in the following table.

Table 4: Summary of connections between the updated FRs and the selected components

| ID of FR | IDs of components, which address the requirements |
|---|---|
| FR1 | C5, C12, C18 |
| FR2 | C14, C15, C16 |
| FR3 | C5, C12, C18 |
| FR4 | C5, C12, C14, C15, C16 |
| FR5 | C5, C12, C18 |
| FR6 | C4 |
| FR7 | C4 |

| | |
|---|---|
| Modified FR8 | C4 |
| FR9 | C4, C19 |
| FR10 | C24 |
| Modified FR11 | C24 |
| Modified FR12 | C24 |
| Modified FR13 | C24 |
| Modified FR14 | C24, C24 |
| Modified FR15 | C24 |
| FR16 | C24 |
| FR17 | C4, C24 |
| FR18 | C18 |
| FR19 | C21 |
| FR20 | C21 |
| FR21 | C18, C21 |
| FR22 | C18, C21 |
| FR23 | C9, C11, C14, C15, C18 |
| FR24 | C19, C21 |
| FR25 | C4, C4, C5, C18, C21 |
| FR26 | C18 |
| FR27 | C1, C2, C5, ~~C7~~, C12 |
| FR28 | C1 |
| FR29 | C1, C5, ~~C7~~, ~~C10~~, C12, ~~C20~~ |
| FR30 | C5, C12, C24 |
| FR31 | C1, C5, C12 |
| FR32 | C5, C12 |
| ~~FR33~~ | ~~C5, C11, C17~~ |
| FR33A | C5, C11, C17 |
| FR33B | C5, C11, C17 |
| FR34 | C5, C12 |
| FR35 | C8 |
| FR36 | C5, C12, ~~C20~~ |
| Modified FR37 | C5, C12, ~~C20~~ |
| Modified FR38 | C1, ~~C10~~, C19, ~~C20~~, C22, C23, C24, C24 |
| FR39 | C1, C19 |
| Modified FR40 | C1, C19 |
| FR41 | C1, C19 |
| FR42 | C1, C19 |
| FR43 | C1, C19, ~~C20~~, C22, C23, C24 |
| FR44 | C1, C5, ~~C10~~, C12, ~~C20~~, C22, C23 |
| NFR1 | C5, C12 |
| NFR2 | C5, C12 |
| NFR3 | C1, C5, C12 |
| NFR4 | C5, C12 |

| | |
|---|---|
| NFR5 | C5, C6, C12 |
| NFR 6 | C1, C2, C3, C8, C22 |

## 5.3 The implementation status of the planned components

The development of MVP.v2 was done in accordance with the requirements previously specified, and with the reviews received from WP5 after the first demonstration phase execution and initial lessons learnt provided.

Currently MVP v2 supports the onboarding of datasets, applications, and services. For the dataset cases, the storage backend is currently restricted to be the platform's CKAN instance. For the services cases, an OPENAPI specification can be uploaded to the platform in the onboarding phase, but it is not acted upon to, for example, configure routing. Likewise, application onboarding is supported, but the corresponding Docker images must be pushed onto the platform's docker registry in a separate operation. Finally, controlled access to applications is currently limited to HTTP GET operations (as per the current limitations in the DSC implementation), although non-controlled (e.g., directly to the application's container) access is also available and can be kept securely within the consumer's network.

The platform interfaces, implemented as customized CKAN instances, are still in the process of development. MVP.v2 brings the metadata management in the platform closer to the requirements specified in the TRUSTS-IM (See D2.7), in as far as explicitly input metadata fields. That is, the different fields in the TRUSTS-IM specified for describing assets have been included into the UI of the platform, along with querying the vocabularies as stored in the Vocabulary Management component. However, missing points in the interface include the extraction of artifact-specific metadata, such as configurations and routing mechanisms, from configuration files uploaded when onboarding applications and services. This functionality is, furthermore, necessary for the automatic deployment of applications acquired through TRUSTS, so that MVPv.2 still requires a manual execution of said applications via the node's Docker daemon. With regards to the user experience of the platform interfaces, a lot of work has been done on the creation of the necessary style sheets to be integrated into the CKAN extensions, but not all of them have been incorporated into the extensions themselves.

With respect to the contract negotiation mechanisms provided by the DSC, these have been incorporated into the platform interfaces by allowing a provider to choose among the types of offers which their asset to be accompanied with, and a consumer to choose from among the different offers available. Implementation is still leaking on more general offer types (currently only those explicitly supported by the DSC are available), on the offer-acceptance workflow (currently offers are automatically accepted), and on the billing mechanism. All of the new (and upcoming) functionalities of the CKAN platform are implemented in the form of extensions, and they have been successfully tested and deployed on the latest up-stream version of CKAN.

The Vocabulary Management component is currently deployed outside of the TRUSTS platform itself, as development is ongoing for allowing it to be deployed in a dockerized environment. In its current state, its operation has been tested and validated using the Jena Fuseki database, which forms the basis of the Metadata Broker metadata storage. This allows for centralization of both metadata and vocabularies and, thus, for joint queries among them. Since the necessary vocabularies are not finalized (as part of Task 3.4), it has not been explicitly included in the MVP.v2, but it has been tested to work with the other components (see above).

With respect to the Metadata Broker, its development (which is not in the scope of this project) is in a mature state and it has been successfully deployed and integrated into the MVPv.2 using the Apache Jena Fuseki backend. The IDSCPv2 interface of the Metadata Broker is actively being used in MVPv.2 for the transmission of metadata, and to enable the federated search of assets within the platform. Finally, the possibility of queries on both the metadata and the accompanying vocabularies simultaneously has been tested, and basic versions of these (so far with a static version of the vocabularies loaded into the triple store) have been tested.

DSC now allows for:

- Setting up some policies in offers (e.g., single access, time restricted access)
- Checking compliance with these (access control)
- CKAN-independent onboarding of assets, by its user-friendly API.

From the deployment point of view, a docker compose file was provided for each of the corporate nodes to launch the components executed therein, after pulling them from the TRUSTS-operated docker registry. The configuration of these nodes is done by adjusting a single file (whose contents are fed into the corresponding containers in the form of environment variables), allowing for a certain amount of customization in the deployment. Apart from downloading the docker-compose file and creating the configuration file, a node operator had to announce to the central node administrator some parameters needed for connecting to it, which would then in turn be broadcast to the other participants.

It is important to note that the node deployment operation will always be very technical in nature, requiring operators to engage their technical personnel for this event. MVP.v2 has reduced the complexity of the deployment operation by removing the need for metadata harvesting, and its corresponding configuration. Further simplification of the deployment process will be undertaken in the next iterations of the platform.

In the following table, the changes of implementation status of all components are listed. Again, the method of highlighting the change using colours is used. Therefore, new components are coloured with green, and the eliminated ones are coloured with red and stroked-through.

Table 5: Summary of the implementation status of the platform components

| Component name | Status | Details |
|---|---|---|
| ~~C1 - Trusted Connector~~<br>C1 - Dataspace Connector | ~~Reused from IDS~~<br>Reused from IDS | ~~IDS Trusted Connector Version 4.0.0~~<br>IDSA Dataspace Connector Version 6.2 |
| C2 - Dataflow Router | Planning and design phase | - |
| C3 - Reverse Proxy | Reused from Open-Source Software | NGINX, Version 1.12.0 |
| C4 - Recommender | Developed by KNOW for TRUSTS | Details are described in D3.12 (M18) |
| C5 - Platform Interface | Reused from Open-Source Software | CKAN (Comprehensive Knowledge Archive Network), Version 2.9.3<br>Source Code in TRUSTS repository:<br>https://gitlab.com/trusts-platform/ckan/-/tree/trusts |
| C6 - Landing Page | Planning and design phase | - |
| ~~C7 - Asset Consumer~~ | ~~Planning and design phase~~ | ~~CKAN, as listed for C5 – Platform Interface might also be used to implement this component.~~ |
| C8 - Notification Service | Planning and design phase | - |
| C9 - Metadata Mapper | Reused from DMA | Source Code in TRUSTS repository: https://gitlab.com/trusts-platform/dma-simple-RMLmapper<br>More details in D3.7 (M18) |
| ~~C10 - Usage Control~~ | ~~Planning and design phase~~ | ~~More details in D3.7 (M18).~~ |

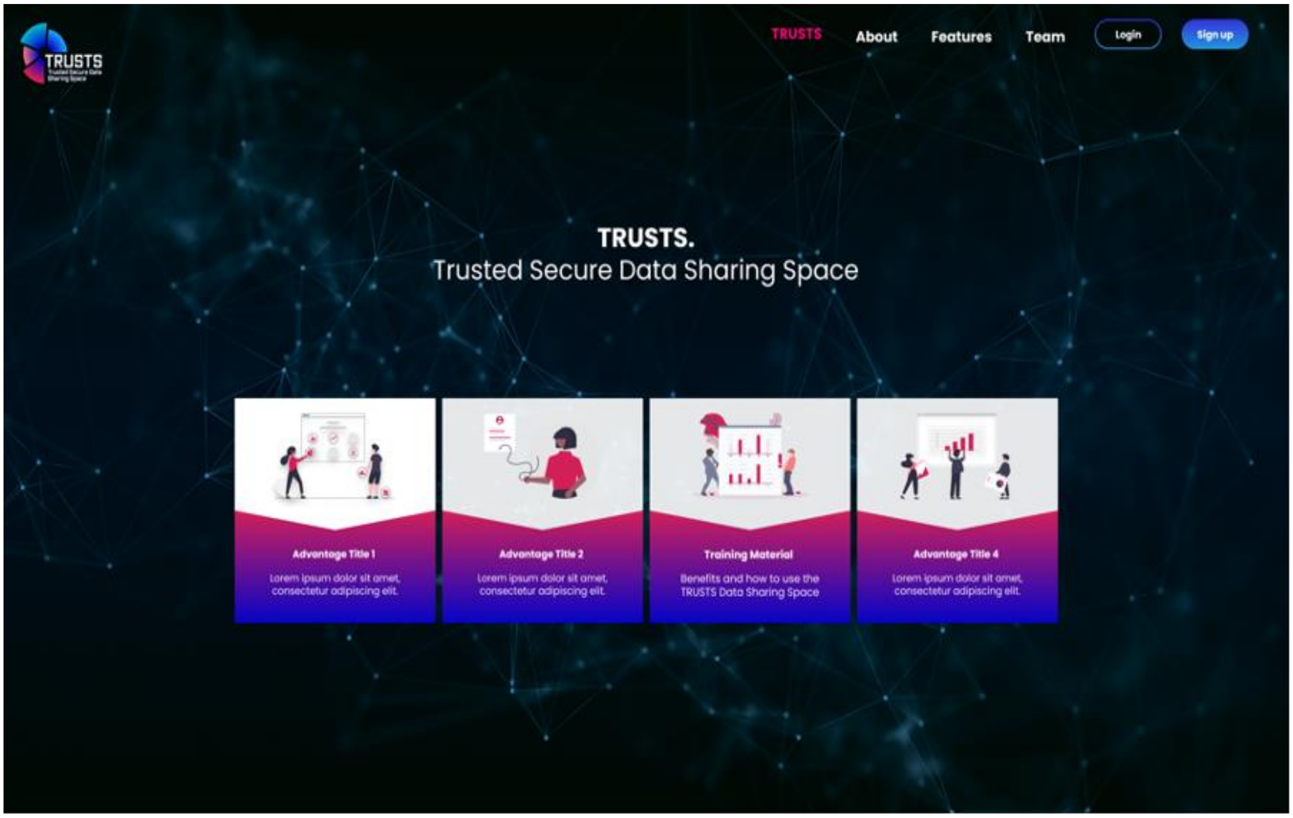
| | | |
|---|---|---|
| C11 - Mapping Builder | Reused from DMA | Source Code in TRUSTS repository: https://gitlab.com/trusts-platform/dma-metadata-mapping-builder |
| C12 - Corporate Interface | Reused from Open-Source Software | CKAN (Comprehensive Knowledge Archive Network), Version 2.9.3<br>Source Code in TRUSTS repository: https://gitlab.com/trusts-platform/ckan/-/tree/trust s |
| ~~C13 - Services Consumer Adapter~~ | ~~Planning and design phase~~ | ~~-~~ |
| C14 - Data Exchange TRUSTS Component | Planning and design phase | This component is deployed in TRUSTS nodes and ingests assets from third-party data-markets and EOSC and its related initiatives into the TRUSTS catalog.<br>More details in D3.4 (M12). |
| C15 - Data Exchange Client Component | Planning and design phase | This component provides an interface to specify and select data assets and to map their metadata schema into a format understood by TRUSTS. The component communicates with the Data Exchange TRUSTS Component.<br>More details in D3.4 (M12). |
| C16 - Registry of Data markets | Planning and design phase | This component lists existing third-party data markets and relevant initiatives of EOSC. It serves as an address book routing the communication between the Data Exchange TRUSTS Component and the Data Exchange Client Components installed on the premises of third-party data markets and EOSC initiatives.<br>More details in D3.4 (M12). |
| C17 - Business Support Services | Planning and design phase | - |
| C18 - Broker + Metadata Storage | Reused from IDS | IDS Metadata Broker + Apache Jena Fuseki<br>Details in D3.7 (M18) |
| C19 - App Store | Planning and design phase | - |
| ~~C20 - Identity Provider + Key Distribution System~~ | ~~Planning and design phase~~ | ~~-~~ |
| C21 - Vocabulary Services | Developed by SWC | PoolParty https://www.poolparty.biz/<br>Details in D3.7 (M18) |

| | | |
|---|---|---|
| ~~C22 - Dynamic Attribute Provisioning System(DAPS)~~<br>C22 - Distributed Authorisation Component | ~~Reused from IDS~~<br>Planning and design phase | ~~Instance of service provided for testing by Fraunhofer~~<br>provided by a combination of the IDS Dynamic Attribute Provisioning System (DAPS which is provided for testing by Fraunhofer) implementation or the Small Step CA. |
| C23 - Automated Certificate Management Environment (ACME) | Reused from IDS | Instance of service provided for testing by Fraunhofer |
| C24 - Smart Contract Execution | Under evaluation regarding the possibility of development by FhG and EMC | More details will be in D3.3 (M36) |

## 5.4 Snapshots for the Landing Page of TRUSTS platform

Before ending this subsection, some of the snapshots of the landing page of TRUSTS are presented to show how non-registered users can enroll and login, reset password, as well as learn about TRUSTS platform, legal terms, contact points, etc.

The users can interact with each of the advantages, which will provide them with information about how the TRUSTS platform will be useful to them. Aside from this information, users also have access to the navigation bar, which allows them to navigate the platform, as shown in Figure 8. While a user is logged out, the navigation options include links to the About page as well as buttons to the "Login" and "Signup" pages.



TRUSTS.
Trusted Secure Data Sharing Space

Advantage Title 1
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Advantage Title 2
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

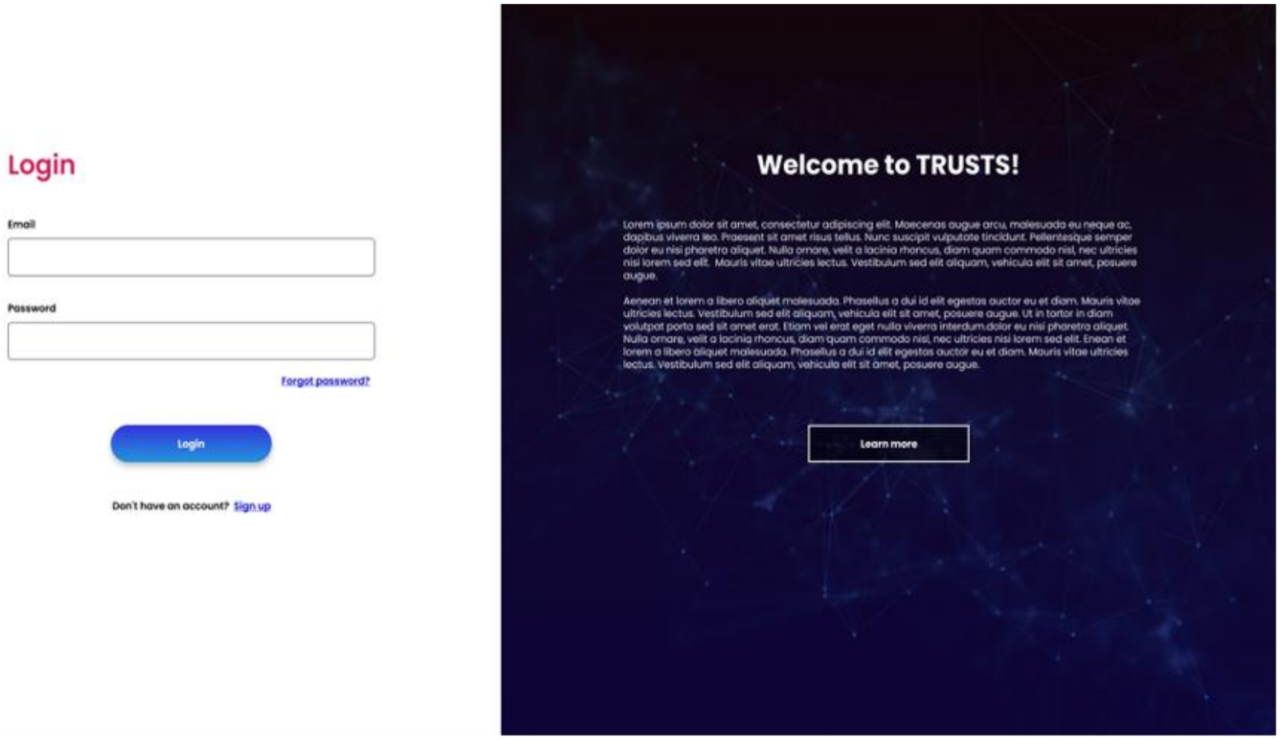
Training Material
Benefits and how to use the TRUSTS Data Sharing Space

Advantage Title 4
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Figure 8: Snapshot of TRUSTS 'Landing Page'

### 5.4.1 Login page

When users select the "Login" button from the navigation bar, they are directed to the corresponding page, as presented in Figure 9. This page comprises the necessary input fields for users to log in, as well as a welcome message to the TRUSTS platform. Users can log in to the platform using their TRUSTS credentials. If they do not have an account, a sign-up link is provided that will direct them to the "Sign up" page. In addition, if they have forgotten their password, a "Forgot password" link is provided for them to be able to change their password.



Welcome to TRUSTS!

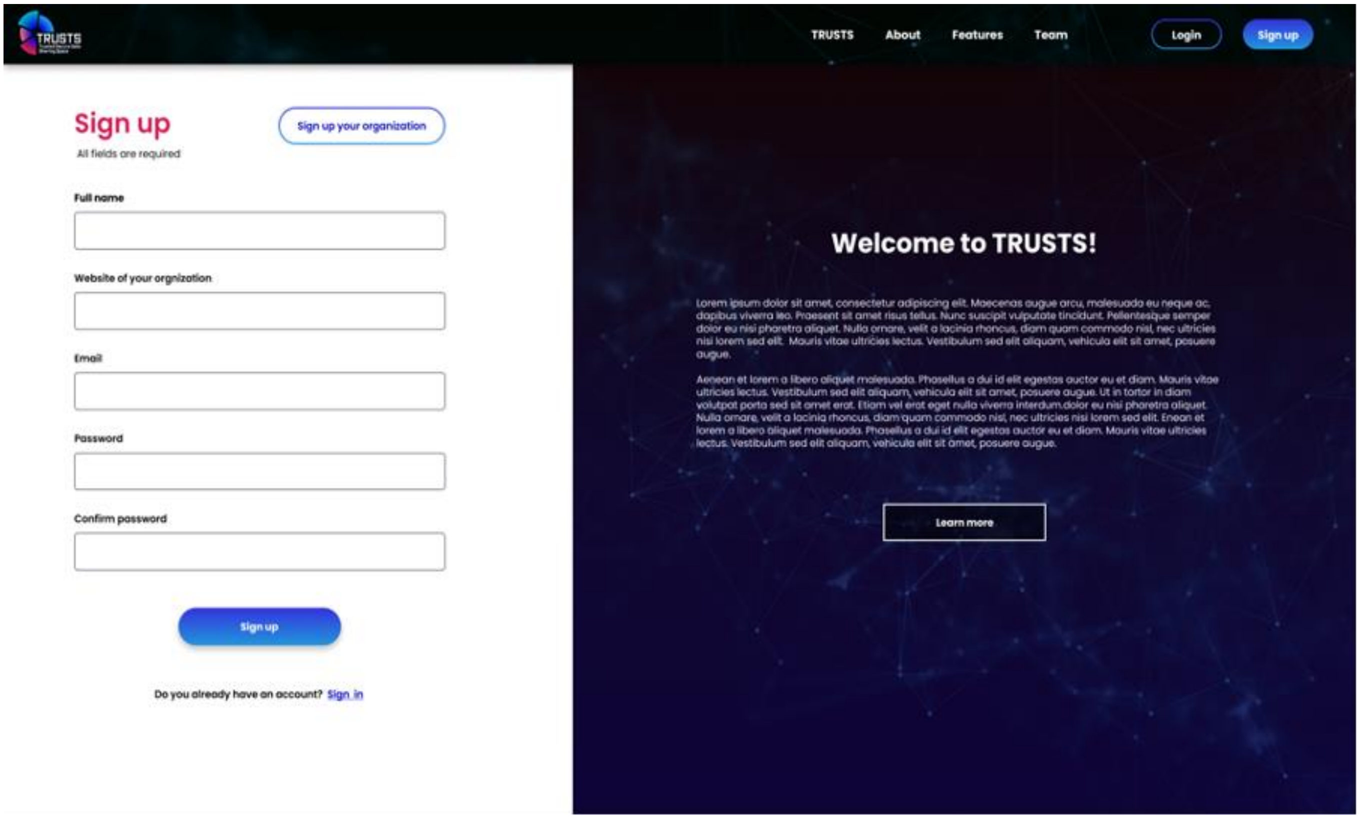
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas augue arcu, malesuada eu neque ac, dapibus viverra leo. Praesent sit amet risus tellus. Nunc suscipit vulputate tincidunt. Pellentesque semper dolor eu nisi pharetra aliquet. Nulla ornare, velit a lacinia rhoncus, diam quam commodo nisl, nec ultricies nisi lorem sed elit.  Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue.

Aenean et lorem a libero aliquet malesuada. Phasellus a dui id elit egestas auctor eu et diam. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue. Ut in tortor in diam volutpat porta sed sit amet erat. Etiam vel erat eget nulla viverra interdum.dolor eu nisi pharetra aliquet. Nulla ornare, velit a lacinia rhoncus, diam quam commodo nisl, nec ultricies nisi lorem sed elit. Enean et lorem a libero aliquet malesuada. Phasellus a dui id elit egestas auctor eu et diam. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue.

Figure 9: Snapshot of TRUSTS ‘Login Page’

### 5.4.2 Sign up page

Figure 10 presents the "Signup" page which can be accessed when users click on a "Sign up" button. This page includes the necessary input fields so that the users can sign up, as well as a welcoming message to the TRUSTS platform. The users can choose to create an account for themselves or for their organization. If they wish to create an account for their organization, they can do so by clicking on the "Signup your organization" button, which is located at the top right of the input form.



Welcome to TRUSTS!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas augue arcu, malesuada eu neque ac dapibus viverra leo. Praesent sit amet risus tellus. Nunc suscipit vulputate tincidunt. Pellentesque semper dolor eu nisi pharetra aliquet. Nulla ornare, velit a lacinia rhoncus, diam quam commodo nisl, nec ultricies nisi lorem sed elit. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue.

Aenean et lorem a libero aliquet malesuada. Phasellus a dui id elit egestas auctor eu et diam. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue. Ut in tortor in diam volutpat porta sed sit amet erat. Etiam vel erat eget nulla viverra interdum.dolor eu nisi pharetra aliquet. Nulla ornare, velit a lacinia rhoncus, diam quam commodo nisl, nec ultricies nisi lorem sed elit. Enean et lorem a libero aliquet malesuada. Phasellus a dui id elit egestas auctor eu et diam. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula elit sit amet, posuere augue.

Figure 10: Snapshot of TRUSTS ‘Signup Page’

### 5.4.3 Organization sign up page

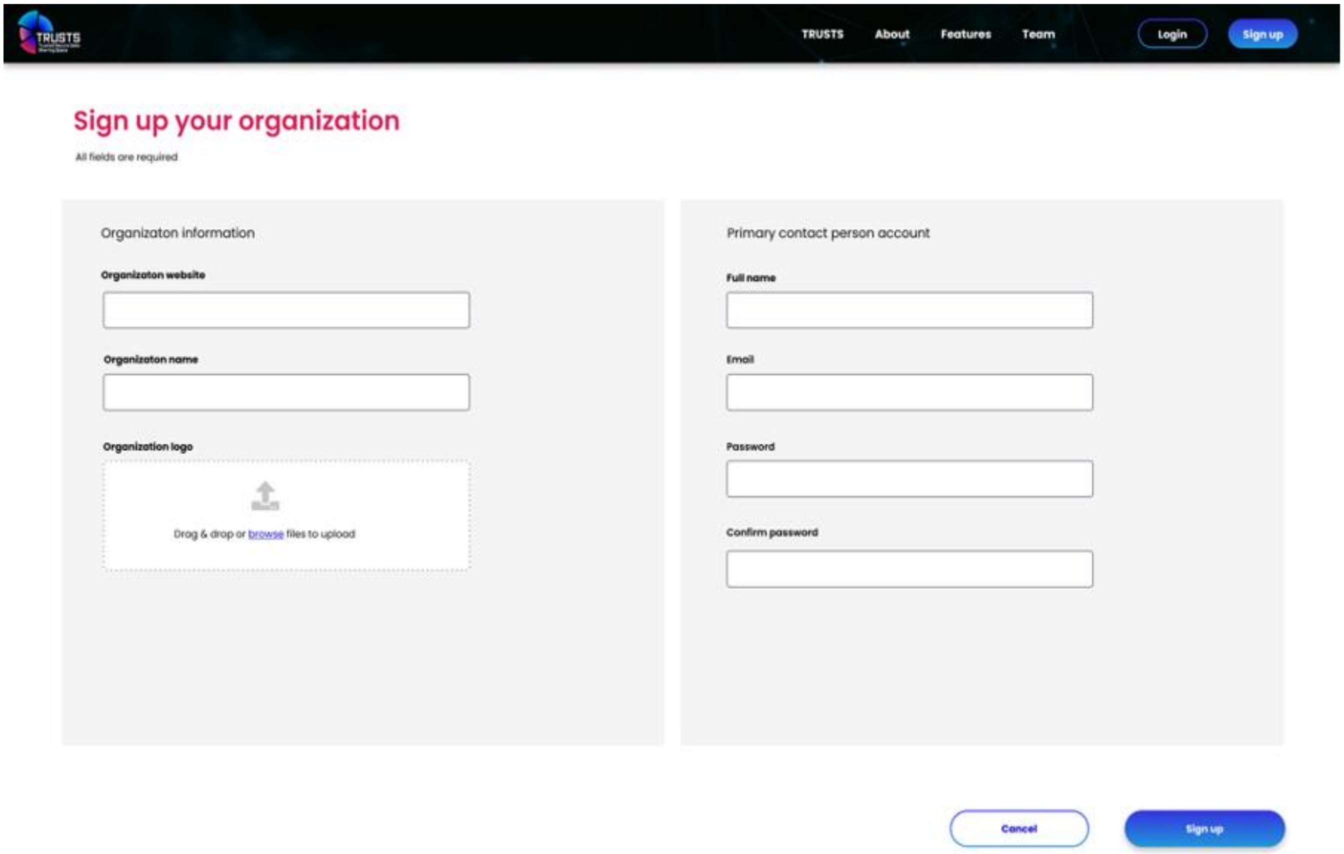
Users can sign up their organization on the Signup your organization page (as shown in Figure 11) by providing the organization's website and name, as well as the organization's logo. In addition to that information, users must also specify the person who will oversee the organization's account. For that person they must provide information such as the full name, email address, and a password. After this information is specified, a new organization is created.

### 5.4.4 About page

From the navigation bar the users have also access to the About page, depicted by Figure 12. This page provides information about the TRUSTS platform and its features, with the goal of capturing the user's attention and increasing the likelihood that they will join the platform.

Specifically, the information displayed in the About page is the following:

- An “About” tab, which contains a brief message and/or description of the TRUSTS platform as well as some of the key features of the platform.
- A “Features” tab, where the key features of the platform are addressed in more detail.
- The “Our team” tab, which displays information and the logo of each partner of the TRUSTS project.
- General information about the Horizon 2020 TRUSTS project and a button leading to the project’s website.



Sign up your organization

All fields are required

Organizaton information

Organizaton website

Organizaton name

Organization logo

Primary contact person account

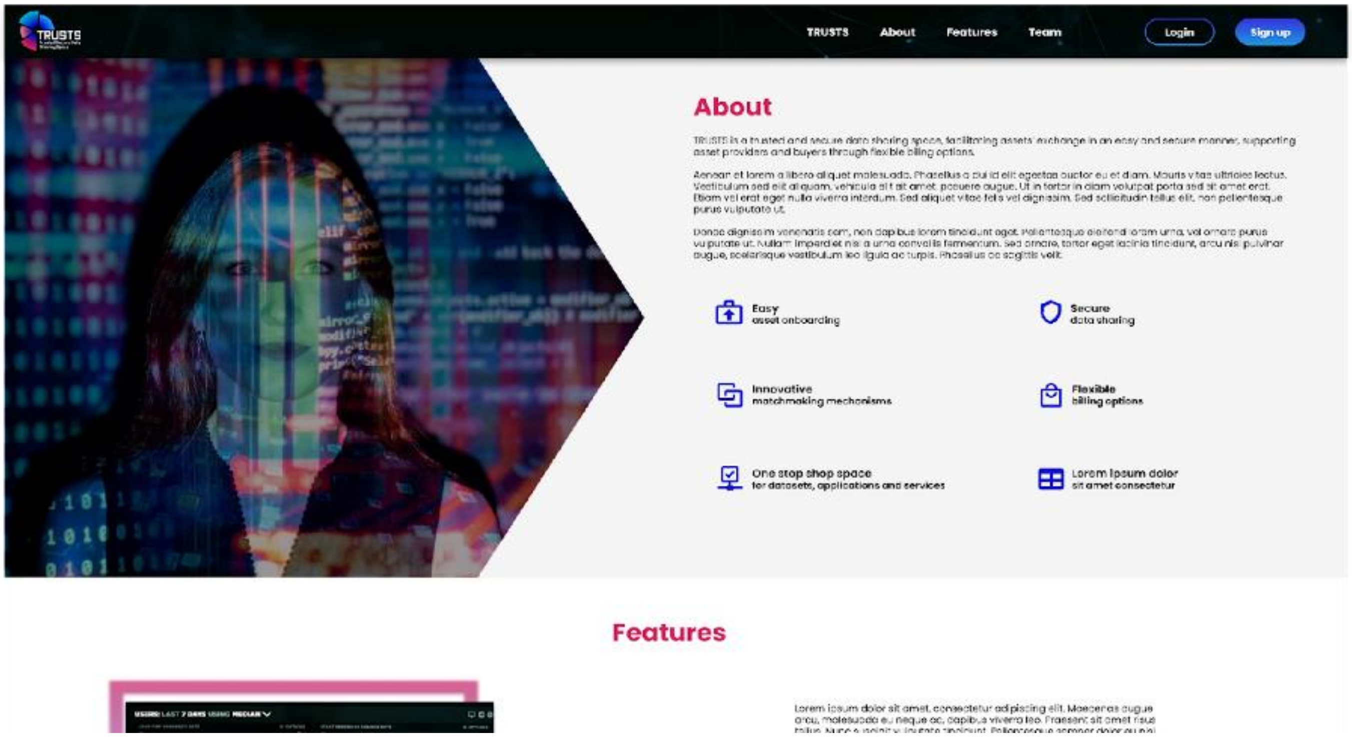
Full name

Email

Password

Confirm password

Figure 11: Snapshot of TRUSTS Organization ‘Signup Page’



About

TRUSTS is a trusted and secure data sharing space, facilitating assets' exchange in an easy and secure manner, supporting asset providers and buyers through flexible billing options.

Aenean et lorem a libero aliquet malesuada. Phasellus a dui id elit egestas auctor eu et diam. Mauris vitae ultricies lectus. Vestibulum sed elit aliquam, vehicula sit at amet, posuere augue. Ut in tortor in diam volutpat porta sed sit amet erat. Etiam vel erat eget nulla viverra interdum. Sed aliquet vitae felis vel dignissim. Sed sollicitudin tellus elit, non pellentesque purus vulputate ut.

Donec dignissim venenatis sem, non dapibus lorem tincidunt eget. Pellentesque eleifend lorem urna, vel ornare purus vulputate ut. Nullam imperdiet nisi a urna convallis fermentum. Sed ornare, tortor eget lacinia tincidunt, arcu nisi pulvinar augue, scelerisque vestibulum leo ligula ac turpis. Phasellus ac sagittis velit.

Easy
asset onboarding

Secure
data sharing

Innovative
matchmaking mechanisms

Flexible
billing options

One stop shop space
for datasets, applications and services

Lorem ipsum dolor
sit amet consectetur

Features

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas augue arcu, malesuada eu neque ac, dapibus viverra leo. Praesent sit amet risus

Figure 12: Snapshot of TRUSTS ‘About Page’

# 6 Conclusion and next actions

The update about the status report of the TRUSTS platform is described in this document. It reports a snapshot of the current platform status.

This is the second of three versions of this deliverable focusing on a second iteration operational product of the platform that covers the FRs previously specified, and with the lessons learnt received from the WP5 first demonstration phase of the platform execution.

The architecture of this platform version is explained based on the technical architecture of the TRUSTS platform described in deliverable D2.7 “Architecture design and technical specifications document II” in parallel submitted, December 2021. Thus, this deliverable is in strong collaboration with other deliverables.

The updated FRs collected from the project participants who are working on tasks related to the implementation of the platform are also summarized in this document, followed by a documentation on the chosen and selected core components. The components were described in terms of their suitability for the updated FRs of the platform to ensure agile development and the implementation of the platform. While the concept of a Minimum Viable Product (MVP) is used to involve relevant stakeholders in the evolution of the platform, at different points in time during the project, the report describes the second iteration TRUSTS platform from an architectural perspective by giving a general overview of the technical architecture as well as the overflow of the main functionalities of this version by showing their sequence diagrams. Finally, snapshots for the landing page are shown.

To explore the next actions of this work, the following version of this deliverable (D3.11), will describe the next iterations of the TRUSTS platform. The next versions of the platform components will be introduced and explained in more detail in D3.11 containing additional component implementations as available experience and feedback will be incorporated into its/their development from the other tasks in WP3, and on interactions with the use cases in WP5.

The technical focus will be on integrating the components into a homogeneous platform. As part of this, it is planned to extend the knowledge about connectors. Another aspect involves setting up a testing environment with instances of all components required for providing a secure data marketplace and secure services. All future iterations of the platform implementations after MVP.v2 will be described in deliverable D3.11, which is the third and final version of the platform status report.

Furthermore, if there will be any changes to the architecture design and technical specifications that have been specified in the deliverable D2.7, then it will be updated in the D3.11 scheduled to be released at the end of the project duration, December 2022 (M36).

# 7 References

[1] M. Traub, et al., "Broker and Assessment Technology Specification and Development Road", Data Market Austria, May 2017.

[2] B. Otto, et al., "Reference Architecture Model Version 3.0", International Data Spaces Association, April 2019.

[3] V. Lenarduzzi, D. Taibi, "MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product", Euromicro SEAA, 2016.

[4] J. Münch, et al. "Creating minimum viable products in industry-academia collaborations." International Conference on Lean Enterprise Software and Systems. Springer, Berlin, Heidelber