



Deliverable 3.4

Release of the data valuation tools and trading engine

H2020-EU-2.1.1: **PIMCity**

Project No. **871370**

Start date of project: 01/12/2019

Duration: 33 months

**Deliverable delivery:** 28/02/2022

Deliverable due date: 28/02/2022



## Document Information

Document Name: **Release of the data valuation tools and trading engine**

**WP3 – Title: Understanding the new data economy**

Tasks 3.2 and 3.3

Revision: 01

Revision Date: 23/02/2022

Authors: UC3M, WIBSON, IMDEA

## Dissemination Level

Project co-funded by the EC within the H2020 Programme		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

## Approvals

	Name	Entity	Date
<b>WP Leader</b>	Rubén Cuevas Rumín	UC3M	11/02/2022
<b>Author</b>	Amir Mehrjoo	UC3M	11/02/2022
<b>Author</b>	Ángel Cuevas Rumín	UC3M	11/02/2022
<b>Author</b>	Pedro Reveriego	UC3M	11/02/2022
<b>Author</b>	Nikolaos Laoutaris	IMDEA	11/02/2022
<b>Author</b>	Santiago Andrés Azcoitia	IMDEA	11/02/2022
<b>Author</b>	Daniel Fernandez	WIBSON	11/02/2022
<b>Author</b>	Rodrigo Irarrazaval	WIBSON	11/02/2022
<b>Author</b>	Javier Calvo	WIBSON	11/02/2022
<b>Coordinator</b>	Marco Mellia	POLITO	11/02/2022
<b>Reviewer</b>	Ioannis Arapakis	TID	23/02/2022



## Document history

Revision	Date	Modification
<b>Version 1</b>	23/02/2022	

## List of abbreviations and acronyms

Abbreviation	Meaning
<b>DVTMP</b>	Data Valuation Tool from Market Perspective
<b>DVTUP</b>	Data Valuation Tool from end User Perspective
<b>TE</b>	Trading Engine
<b>DVT</b>	Data Valuation Tool
<b>CPM</b>	Cost Per Mille
<b>CPC</b>	Cost Per Click
<b>WTP</b>	Willingness-to-pay
<b>WTA</b>	Willingness-to-accept
<b>WP</b>	Work Package



## Executive Summary

This deliverable describes the final releases of the tools devoted to understanding the value of users' data (Data Valuation Tools) and the tool for trading users' data with data buying entities (Trading Engine). It also provides the final software release of the data valuation tools and the trading engine, along with a user guideline for utilizing the different tools (T3.2, T3.3). In particular, the *Data Valuation Tools from the Market Perspective (DVTMP)*, the *Data Valuation Tools from User Perspective (DVTUP)* and the *Trading Engine (TE)*.

This document is part of the PIMCity Project, funded from the Horizon 2020 Program (ICT-13-2018-2019) under Grant Agreement number 871370.

The technical decisions and design choices have been carefully discussed in plenary meetings as well as specific meetings of WP3 members. Based on these decisions, the partner responsible for the development of each tool has taken care of developing, implementing the tools. The final version of the tools is available on the project's [online repository](#). This document presents the user guidelines for each tool.



# Index

<b>1</b>	<b><i>Introduction</i></b> .....	<b>6</b>
<b>2</b>	<b><i>Data Valuation Tools from Market Perspective</i></b> .....	<b>8</b>
<b>2.1</b>	<b><i>Introduction</i></b> .....	<b>8</b>
<b>2.2</b>	<b><i>Technical Design</i></b> .....	<b>8</b>
<b>2.3</b>	<b><i>Installation</i></b> .....	<b>9</b>
<b>2.4</b>	<b><i>Execution</i></b> .....	<b>10</b>
	<b><i>2.4.1 Configuration</i></b> .....	<b>10</b>
	<b><i>2.4.2 Usage Examples</i></b> .....	<b>10</b>
<b>2.5</b>	<b><i>Changes</i></b> .....	<b>11</b>
<b>2.6</b>	<b><i>License</i></b> .....	<b>11</b>
<b>3</b>	<b><i>Data Valuation Tool from the User's Perspective</i></b> .....	<b>11</b>
<b>3.1</b>	<b><i>Introduction</i></b> .....	<b>11</b>
<b>3.2</b>	<b><i>Installation</i></b> .....	<b>12</b>
<b>3.3</b>	<b><i>Usage</i></b> .....	<b>13</b>
	<b><i>3.3.1 Execution</i></b> .....	<b>13</b>
	<b><i>3.3.2 Usage Examples</i></b> .....	<b>13</b>
<b>3.4</b>	<b><i>Changes</i></b> .....	<b>14</b>
<b>3.5</b>	<b><i>License</i></b> .....	<b>14</b>
<b>4</b>	<b><i>Data Trading Engine</i></b> .....	<b>15</b>
<b>4.1</b>	<b><i>Introduction</i></b> .....	<b>15</b>
<b>4.2</b>	<b><i>Installation</i></b> .....	<b>16</b>
<b>5</b>	<b><i>Usage</i></b> .....	<b>17</b>
<b>5.1</b>	<b><i>Execution</i></b> .....	<b>17</b>
	<b><i>5.1.1 Configuration</i></b> .....	<b>17</b>
	<b><i>5.1.2 Usage Examples</i></b> .....	<b>17</b>
<b>5.2</b>	<b><i>Changes</i></b> .....	<b>17</b>
	<b><i>5.2.1 Additions</i></b> .....	<b>17</b>
<b>5.3</b>	<b><i>License</i></b> .....	<b>17</b>
<b>6</b>	<b><i>Conclusion</i></b> .....	<b>18</b>



## 1 Introduction

This deliverable aims at presenting the final version of the design of the tools developed as part of the Tasks 3.2 and 3.3 within the WP3. It also delivers the final version of the software implementing such tools, namely:

- *Data Valuation Tools from a market perspective (DVTMP)*
- *Data Valuation Tools from an end-user perspective (DVTUP)*
- *An open Trading Engine (TE)*

The TE is considered a key component of a PIM, responsible for trading the data of users registered/handled by the PIM with interested buyers. Hence, the TE serves as a communication interface between the PIM backend and the data buyers. There is a myriad of data types that can be sold. For coherence and concreteness in PIMCity, and the design of the tools referred to in this deliverable, we will focus on the two most common types of data being sold: 1) Bulk data and 2) Real-time audience data.

Bulk data is typically bought in a non-real-time manner to receive information from a (typically large) group of users. Some examples include: a health insurance company may be interested in people's medical records; a car insurance company may be interested in the people's mobility data to understand who drives through tough roads or at higher speeds; a mortgage-issuing company may be interested in people's financial records, etc. None of these data need to be traded in real-time, and typically data buyers try to buy it in bulk. Due to the relevance of the selling of bulk data for a large number of businesses, the TE will be designed and developed to enable the trading of bulk data to PIMs implementing it.

An audience is a term used in marketing to refer to a specific group of the population defined by three parameters: location (where the user is located); demographic information (age, gender, etc.); interests (a list of interests, e.g., outdoor activities, sports, science, automobile, etc.). Most of these parameters are typically extracted from well-known taxonomies such as the one offered by IAB (which is a de-facto standard in digital marketing)<sup>[1]</sup> Online advertising (a.k.a. digital marketing) is arguably one of the most important businesses exploiting data utilization, specifically audience data, to deliver targeted ads to users in real-time. Due to the importance of this business, the TE will be designed and developed to allow PIMs to implement it to trade audiences' data.

One of the main problems that PIMs face, and the fundamental reason for the failure of some of them so far, is their inability to assess the value of data. To address this issue in Task 3.2, we will work on developing two types of Data Valuation Tools (DVTs). In the context of the PIMCity architecture, the DVTs are meant to assist the TE in estimating the value of the data the TE is trading. As indicated above, WP3 will develop two types of DVTs discussed in what follows.

On the one hand, the DVTMP will obtain audiences' value from some of the most important online advertising platforms such as Facebook, Instagram, or LinkedIn. The information



collected by the DVTMP will assist the TE to estimate the value of audiences based on the market price of these on the aforementioned advertising platforms, so that the trading price is well aligned with the market price of the audience. This enables the PIM to implement a combination of the DVTMP and TE to compete in the market of audiences purchasing/selling (e.g., the online advertising market). On the other hand, the DVTUP will analyze the value of data of individual users within a large pool of users. It will apply a value-based fairness principle, which is a novel principle to be used in the context of the data economy. Each user data contributes different data with a relative value compared to the data provided by other users. Let us consider the next example: A company willing to predict the traffic in a city. Professional drivers from the city can share their GPS data so that the aggregate data from a large pool of drivers will help modelling the city's traffic. However, the data sample provided by each professional driver does not have equal value. The driver providing data for the most crowded area, where many other drivers are also providing data, is not of much value since he provides redundant information. However, a driver providing data from less crowded areas, where few other drivers provide data, is much more valuable. Armed with this novel fairness-based data valuation, the DVTUP will enable the TE to estimate the relative value of different data sources in the context of bulk data trading for specific purposes.

Hence, following the modular design concept of PIMCity that will allow a PIM to use the whole PIMCity architecture or to implement just a few of its modules, WP3 implements two independent DVT modules which address the two considered data trading scenarios: <DVTUP – bulk data purchase scenario> and <DVTMP – audience-based purchase scenario>. Based on this design concept, integrating the two DVTs will be done independently with the TE.

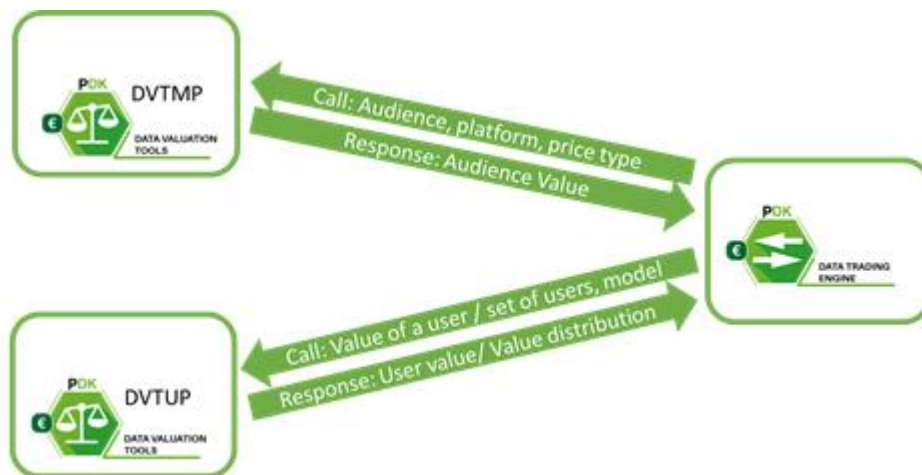


Figure 1 Summary of modules and their interactions

**Figure 1** shows a scheme for the high-level integrated design of the tools covered by this deliverable. The final design and the first version of the software implementing each of the three mentioned modules (DVTMP, DVTUP, and TE) will be described in the rest of the document. In addition, the details of the integration of the DVTMP-TE and DVTUP-TE and the first version of the APIs implementing it will also be presented.



Note that these modules' design and implementation has been done taking into account the requirements detailed in D1.1 from PIMCity. We suggest the reader interested in knowing the details, to refer to such deliverable. This document presents the final design of the technical modules from WP3.

<https://iabtechlab.com/standards/audience-taxonomy/>

## 2 Data Valuation Tools from Market Perspective

Data valuation tools provide an estimation of the audiences' monetary value for advertising in online social media platforms (Facebook, instagram and linkedin).

### 2.1 Introduction

The Data Valuation Tools from the market perspective (DVTMP) module developed in PIMCity is leveraging some of the most popular existing online advertising platforms to estimate the value of hundreds to thousands of audiences. The DVTMP module aims to provide the monetary value of audiences traded on the main online advertising platforms. This serves any PIM deciding to implement the DVTMP module to have a realistic estimation of audiences' value to be traded. Since the information about values collected from these advertising platforms is based on aggregated historical pricing data, we can assert that DVTMP provides full-privacy guarantees. Moreover, a given audience's value can be obtained in real-time from the referred online advertising platform.

### 2.2 Technical Design

The flow chart shown below summarizes the conceptual execution process of the DVTMP module for a specific advertising platform. Note that there is an independent submodule taking care of each advertising platform. This design provides flexibility to easily extend the DVTMP to include new data sources from other advertising platforms. The module communicates using JSON format API provided in OpenAPI format in the [project repository](#). The module first identifies the marketing platform targeted in the request. Then, the corresponding platform submodule analyses the request's parameters. Since each marketing platform has its peculiarities regarding the offered information and how to access it (e.g., search for ids, audiences, or prices), their submodules also independently.



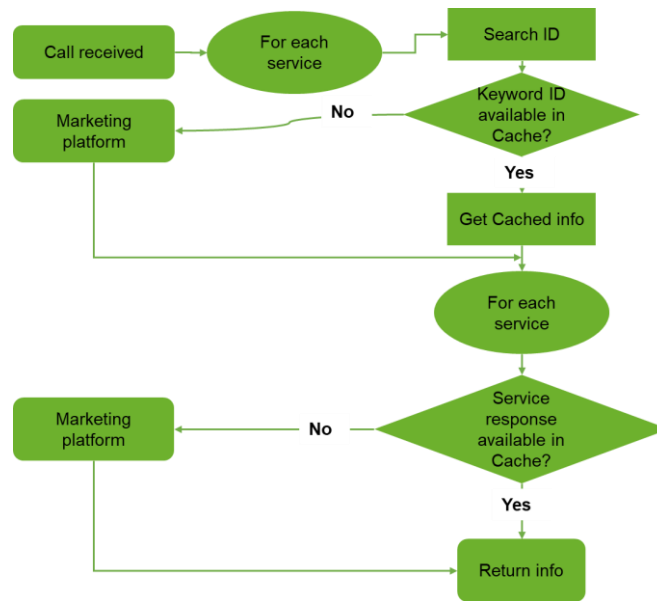


Figure 1 Flowchart of each platform submodule

## 2.3 Installation

The following steps should be taken to install the stand-alone module:

- 1- Download and install Docker Desktop using the following link address <https://www.docker.com/products/docker-desktop>.
  - Windows users should primarily install the Linux subsystem (WSL) to use Docker Desktop. The instructions are available in the following link address: <https://docs.microsoft.com/en-us/windows/wsl/install>
- 2- Download (Clone) DVTMP from the project repository using the following link address: <https://gitlab.com/pimcity/wp3/dvtmp>
- 3- Open a terminal/shell/cmd in the module home folder directory
- 4- Run the following commands in the terminal:

```
$ cd /Build/request_handler
```
- 5- Run the following command to run the docker container. This command creates a local server at port 3700.

```
$ docker-compose up
```

  - Linux users should run this command using sudo permission ( `$ sudo docker-compose up` )



## 2.4 Execution

### 2.4.1 Configuration

Defining the environment variables in a ".env" file at the root directory of the folder with the PDK deployed.

### 2.4.2 Usage Examples

The installed module communicates using POST API sent to the server located. Following is an example of a query to the server getting the estimated prices in CPM (Cost Per Million Ads. impression) for targeting "female," between 18 to 50 years old, interested in "Travel" who live in "Italy" in the Facebook platform.

Using Python:

The server is located on port 3700 by default and this address can be changed in "dvtmp/Build/request\_handler/docker-compose.yml".

The server receives a POST request with an audience description and responses with a value estimation using Facebook, Instagram and LinkedIn platforms. The following lines of code ask for an estimation of the value, expressed in Cost Per Mille impressions, of an audience in Instagram composed of women between 18 and 50 living in Italy and interested in travelling:

```
import json
import requests

url = '127.0.0.1:3700/auValuator'

body = {
    "min_age" : 18,
    "max_age": 50,
    "interest": ["Travel"],
    "location": "IT",
    "platform": "instagram",
    "priceType": "CPM",
    "gender": "female"
}

r=requests.post(url, data=json.dumps(body))
```

The response should then look like this:

```
{
  "status": "success",
  "data": {
    "audience_validated": {
      "max_age": 50,
      "min_age": 18,
```



```
    "interest": [
      "Travel"
    ],
    "location": "IT"
  },
  "Platform": "instagram",
  "info": {
    "title": "Data Valuation tool Market Perspective",
    "version": "1.1.0"
  },
  "tags": {
    "name": "auValuator",
    "description": "realtime value"
  },
  "value": 2.276400888218196,
  "currnecy": "USD"
}
```

## 2.5 Changes

The DVTMP in the new update has implemented the authorization controller module using [Keycloak](#) for communication with the other modules. On top of that, the module can now provide estimated prices for advertising on the LinkedIn platform.

## 2.6 License

The DVTMP is distributed under AGPL-3.0-only, see the LICENSE file.

DVTMP Copyright (C) 2021 UC3M

# 3 Data Valuation Tool from the User's Perspective

## 3.1 Introduction

The Data Valuation Tool from an End-User Perspective (DVTUP) module is meant to provide estimated valuations of end-users' data, for datasets sold through the marketplace as bulk data. In particular, DVTUP will provide tools for the TE to:

1. Provide buyers with a hint of how valuable a piece of data is for a certain type of model or even for a specific task.
2. Calculate a fair breakdown of data transaction charges by seller, looking forward to rewarding each user proportionally to the value that each piece of data from different sellers brings to the buyer for a specific task.



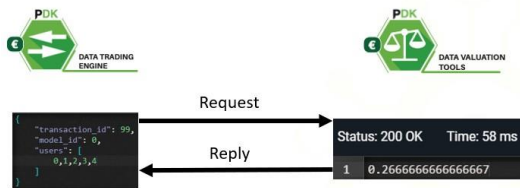
In the first case, the output will be the expected accuracy the buyer would get from a specific dataset of the marketplace.

In the second case, the output will estimate the percentage of a transaction value that corresponds to each seller, and a log of data and results obtained to justify rewards paid to different sellers. For that purpose, different methodologies and algorithms will be designed and implemented to allow data marketplaces to calculate such breakdown in different ways, namely:

- Functionality 1: using traditional heuristics such as data volume or the number of sources See the paper [Try Before You Buy: A practical data purchasing algorithm for real-world data marketplaces](#),
- Functionality 2: using a value-based data evaluation framework to test data on the specific task the buyer is intended to use it in, and breaking the payment according to a value function (e.g., accuracy yielded by a given model or algorithm) provided by the buyer. For more information, please see the paper [Computing the Relative Value of Spatio-Temporal Data in Wholesale and Retail Data Marketplaces](#).

Functionality 1:

Let buyers test the data for their specific task, leading to a close to optimal purchasing process performance



Functionality 2:

Let the TE reward users according to the value they bring to the specific task once a transaction is closed

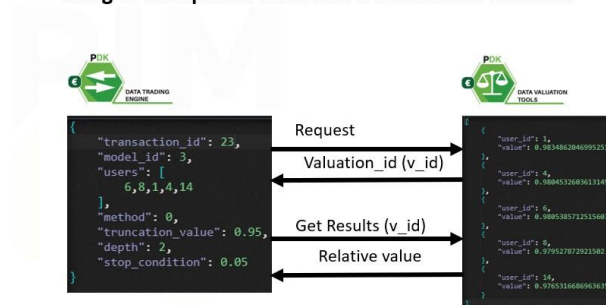


Figure 2 DVTUP interface

An introductory video of the DVTUP can be found [here](#).

### 3.2 Installation

The DVTUP module is implemented in Python3, and requires that the following Python libraries are installed:

- Numpy version 1.19.1 or above - link to Numpy installation instructions [here](#).
- Pandas version 1.1.1 or above - see specific installation instructions [here](#).
- Dask version 2.25.0 or above - link to installation instructions accessible [here](#).
- Flask version 1.1.2 or above - link to installation instructions for Flask accessible [here](#).
- Statsmodels version 0.12.1 or above - see specific installation instructions [here](#).



- MarkupSafe version 1.1.1 or above - see specific installation instructions [here](#).

All the aforementioned libraries are available to be installed by using pip:

```
pip install <library_name>
```

For example, the following set of commands will install the last version of the libraries using pip:

```
$ pip install numpy
```

```
$ pip install pandas
```

```
$ pip install dask
```

```
$ pip install Flask
```

```
$ pip install statsmodels
```

```
$ pip install -U MarkupSafe
```

## 3.3 Usage

### 3.3.1 Execution

To run the server, the following command must be executed from the folder where the source code is stored:

```
python DVTUPServer.py -p <PORT_NUMBER>
```

Where <PORT\_NUMBER> specifies the port number the server will listen to. By default, DVTUP server will use port 5000. Consequently:

```
python DVTUPserver.py - starts a DVTUP server in localhost:5000
```

```
python DVTUPserver.py -p 5005 - starts a DVTUP server in localhost:5005
```

The server will check any dependencies from external PIMCITY modules during the initialization process and will prompt the user in case further action is needed. In this version, the module will work with preloaded models and data to show the feasibility of executing these valuations. Refer to the online documentation to test API calls to the server and use its functionality.

### 3.3.2 Usage Examples

The DVTUP API documentation is available on an online interactive documentation website for the services it provides (See [Swagger Open API definition](#)). By visiting it, the reader can quickly understand and try out the API functionalities. It includes more detail, such as JSON schemas for inputs and outputs of all endpoints and services.

Please refer to the following Jupyter notebooks, which include examples on how to deal with the different APIs and components of the DVTUP.



- `vbde Sample Test.ipynb` - demonstrates how to work with the value-based data valuation framework and their abstract classes, and how to calculate the relative value of data sources in different sample cases.
- `DVTUP Client.ipynb` demonstrates how to start the server and use a REST HTTP client to send valuation requests and get their results.

### 3.4 Changes

- *April 2021* - Initial commit intended to be used in a local environment
- *July 2021* - Modifications and adaptations for the DVTUP to be deployed on PIMCITY's development servers. Debugging minor errors in valuation functions.
- *October 2021* - Removed an error in `vbdeFramework.py` with the toy models that do not use the `vS` buffering of the framework
- *December 2021* - Integration with KeyCloak

### 3.5 License

The DVTUP is distributed under AGPL-3.0-only, see the LICENSE file. This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, see <http://www.gnu.org/licenses>.

Copyright (C) 2021 IMDEA Networks



## 4 Data Trading Engine

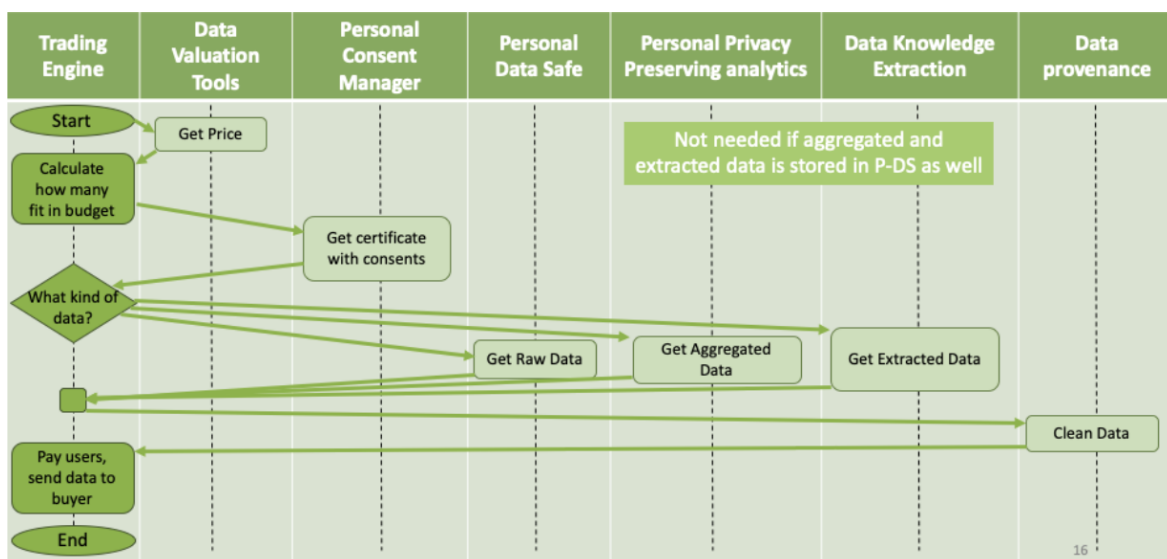
### 4.1 Introduction

The Data Trading Engine (DTE) is responsible for trading the data of users registered/handled by the PIM with interested buyers. Hence, the DTE serves as a communication interface between the PIM backend and the data buyers. There is a myriad of data types that can be sold. The DTE focuses on bulk data and audience data.

Bulk data is typically bought in a non-real-time manner to receive information from a (typically large) group of users. Some examples include: a health insurance company may be interested in people's medical records; a car insurance company may be interested in the people's mobility data to understand who drives through tough roads or at higher speeds; a mortgage-issuing company may be interested in people's financial records, etc. None of these data need to be traded in real-time, and typically data buyers try to buy it in bulk.

An audience is a term used in marketing to refer to a specific group of the population defined by three parameters: location (where the user is located); demographic information (age, gender, etc.); interests (a list of interests, e.g., outdoor activities, sports, science, automobile, etc.). Most of these parameters are typically extracted from well-known taxonomies such as the one offered by IAB (which is a de-facto standard in digital marketing). Online advertising (a.k.a. digital marketing) is arguably one of the most important businesses exploiting data utilization, specifically audience data, to deliver targeted ads to users in real-time.

The DTE is exposed as a REST API which can be accessed by other components in PIMCity or external players like Data Buyers and individuals. The creation of a transaction is depicted in the figure below.





## 4.2 Installation

The documentation and the following instructions refer to a Linux environment, running with **Docker Engine v20.10.x** and **Docker Compose: v1.27.x**. The DTE project has been cloned from [this GitLab repository](#).

Follow accurately the next steps to quickly set-up the DTE backbone on your server. All relevant steps are designed for a Linux machine, perform the equivalent procedure with other environments.

1. Prepare the environment:

```
{
  "status": "success",
  "data": {
    "audience_validated": {
      "max_age": 50,
      "min_age": 18,
      "interest": [
        "Travel"
      ],
      "location": "IT"
    },
    "Platform": "instagram",
    "info": {
      "title": "Data Valuation tool Market Perspective",
      "version": "1.1.0"
    },
    "tags": {
      "name": "auValuator",
      "description": "realtime value"
    },
    "value": 2.276400888218196,
    "currnecy": "USD"
  }
}
```

2. Import the project from the GIT repository:

```
$ git clone https://gitlab.com/pimcity/wp3/datatradingengine.git
```





## 5 Usage

### 5.1 Execution

To run the Trading Engine service, the following command should be executed at the root of the repository:

```
$ docker-compose up
```

#### 5.1.1 Configuration

The Trading Engine can be configured in three similar ways:

Using environment variables.

Defining those environment variables in a file called “.env” at the root directory of the folder with the PDK deployed, declaring them in the same way a variable is defined in the UNIX shell. You can check [this example](#).

Modifying [docker-compose.yml](#) file.

#### 5.1.2 Usage Examples

You can check [here](#) the Swagger OpenAPI definition to see how the API is defined and used. Examples are provided as well.

## 5.2 Changes

### 5.2.1 Additions

- An endpoint to log transactions and credits that were originated in other PDKs. It requires an API Key.
- An endpoint to read transactions that were originated in other PDKs. It requires an API Key.

## 5.3 License

The DTE is distributed under AGPL-3.0-only, see the [LICENSE](#) file.

Copyright (C) 2021 Wibson - Daniel Fernandez, Rodrigo Irarrazaval



## 6 Conclusion

This document described the user guide for final software release the PIMCity modules responsible for estimating the value of the users' data and the module responsible for trading the users' data with interested data buyers. More specifically, this deliverable describes the configuration and the example usage of each module as well as its communication interfaces. The modules whose first complete design has been described in this document are:

- Data Valuation Tools from Market Perspective (DVTMP), from Task 3.2, aims at estimating the value of audiences from different advertising platforms.
- Data Valuation Tools from User Perspective (DVTUP), from Task 3.2., aims at estimating the value of end-users' data using the value-based fairness principle.
- Trading Engine (TE), from Task 3.2, a key piece in the PIMCity architecture responsible for trading end-user's data with interested third party buyers.

The implication and development of these modules has been done collaboratively between the WP3 members, guaranteeing an easy integration. The different WP3 partners have carefully reviewed the document. During the testing and implementation process, some of the presented design decisions may be subject to revision and changes.