This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 883075.

A Platform for Monitoring and Prediction of Social Impact and Acceptability of Modern Border Control Technology

Deliverable D4.3

METICOS Common Information Data Exchange & Storage Model

(first version)

Editor(s):	Radu Popescu
Responsible Partner:	SIMAVI
Status-Version:	Final 1.0
Date:	31/01/2022
Distribution level (CO, PU):	Public

Project Number:	GA 883075
Project Title:	METICOS

Title of Deliverable:	Data Exchange & Storage Model (first version) (first version)
Due Date of Delivery to the EC:	31/01/2022

Workpackage responsible for the Deliverable:	WP4 - Domain Analysis, Requirements Engineering & Specifications Definition
Editor(s):	SIMAVI
Contributor(s):	Georgia Gkioka (ICCS) Radu Gaza (SIM)
Reviewer(s):	ITPFT PPA
Approved by:	All Partners
Recommended/mandatory readers:	WP4

Abstract:	First version of report on METICOS common information data exchange & storage model.
Keyword List:	Common information, data exchange, data model, hierarchical data format
Licensing information:	N/A
Disclaimer	This deliverable reflects only the author's views and views and the Commission is not responsible for any use that may be made of the information contained therein

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.1	12.11.2021	First version	Radu Gaza (SIM)
0.2	14.01.2022	Contributions regarding the data model	Georgia Gkioka (ICCS)
0.3	18.01.2022	Contributions regarding the data interoperability	Radu Popescu (SIM)
0.90	26.01.2022	Contributions	Radu Popescu (SIM)
1	31.01.2022	Final Version 1	Radu Popescu (SIM)

Table of Contents

List of Figures

List of Tables

No table of figures entries found.

Terms and abbreviations

API	Application Programming Interface
BCP	Border Control Point
CDME	Common Data Management Engine
DTVA	Data, Text & Visual Analytics
EC	European Commission
GUI	Graphical User Interface
HDF	Hierarchical Data Format
NGSI	Next Generation Service Interface
SBCT	Smart Border Control Technologies
WP	Work Package

Executive Summary

METICOS Common Data Management Engine (CDME) is responsible for storing the static and the real-time dynamic information acquired from various heterogeneous sources in the METICOS Ecosystem in a uniform way.

This document presents the results of the first iteration of CDME. This covers the technological decisions for implementing CDME and the installation of the products used for implementing it. The final version of CDME and the integration with the other METICOS components will be presented in deliverable D4.6 due in M30.

This deliverable is based on *D4.4 METICOS System Architecture (first version)*, that describes the first version of the METICOS architecture. It is also based on work from *WP7 Big Data Collection, Analysis & Requirements*, more specifically on task *T7.2 Development of Data Models*.

Deliverable D4.3 presents the data model used to organize the various and heterogenic information captured by METICOS platform. Based on this model, it discusses interoperability aspects of the METICOS components.

The deliverable presents the products chosen to store and manage the model and offers some details about their installation. There is also a description of the products used to secure the access to model through authentication and authorisation.

1 Introduction

1.1 About this deliverable

Common Information Data Management Engine (CDME) is the METICOS components that organizes the data captured by the system in an uniform manner, in order to be accessible to all the METICOS components. The data is organized in a hierarchical way, using a common vocabulary. CDME si feed with data from BCPs but also with the output from the METICOS components. The access to CDME will be done through METICOS middleware. This deliverable presents the work done so far on this component.

Some aspects presented in DoA are no longer part of CDME due to change in the project scope. CDME won't manage financial information nor media information. Part of the media information that was planned to be stored in CDME was "credibility profiles for media sources". This implied some ethical concerns regarding the assessment of credibility. After the project scope changes, this is no longer an issue.

1.2 Document structure

Section *2 Data model* describes the data model that will be used to store the information gathered by the component. Section *3 Data Interoperability* discusses how the METICOS components access and use the common data stored by CDME. Section *4 Data storage* offers details about the technological components used to implement CDME and section *5 Data access protocols and access* presents the components that enable the security of the component. Section *6 Conclusions* offers some conclusions.

2 Data model

2.1 Overview

METICOS data model aims to capture the data needed to assess (in both an objective and subjective way) the Smart Border Control Technologies (SBCT) in a harmonized manner. The main entities identified include the type of technology, the subjective feedback and objective performance indicators as the influencing factors, as well as the various types of users of the technologies. Moreover, METICOS data model will allow the assessment of the acceptance of SBCT in other settings, configurations and Border Control Points (BCPs) under the condition that the same data entities and their attributes are captured.

The development of METICOS Data Model is the focus of the work conducted within WP7 and more specifically task T7.2. More specifically, the goal of this task is the design and development of data models for the effective representation of the heterogeneous data gathered for METICOS. In order to achieve the above objectives, we have followed a hybrid approach, both data-driven and knowledge-based. Regarding the data-driven approach, after creating a corpus of relevant literature, we employ Natural Language Processing techniques in order to capture terms and relationships between them which could be used in the construction of the METICOS data model. Then, experts' knowledge is harnessed to confirm the obtained results and build the data model entities, their attributes, metadata and the relationships amongst them, whilst enriching the terms with additional ones if needed. An example of this domain expert knowledge which was taken into account when developing this data model is the METICOS Technology Acceptance Model, which aims to model the factors having an impact on the acceptance of SBCT. The output of this hybrid approach is a data model, which leverages the domain expert knowledge as well as data-driven knowledge extraction and combines the complementary advantages of both approaches. It should be noted that the process of developing the METICOS Data Model is based on an agile approach, and is conducted iteratively. More specifically, after having developed a first version of the data model, this was dispatched for feedback to the technical partners of METICOS. As soon as this input was gathered, the suggested changes and additions were integrated and feedback was requested from technical experts of the field.

More information regarding the data model, the approach which was followed for its implementation as well as a detailed description regarding the entities and their respective attributes, metadata and relationships are going to be presented in Deliverable D7.2. This deliverable will describe also the approaches used to mitigated the biases related to the use of NLP and expert knowledge in the development of the model.

2.2 A High-Level Schema of the Data Model

Below, the high-level concepts of METICOS Data Model are depicted. Please note that, each entity has additional attributes which are not shown in the schema.

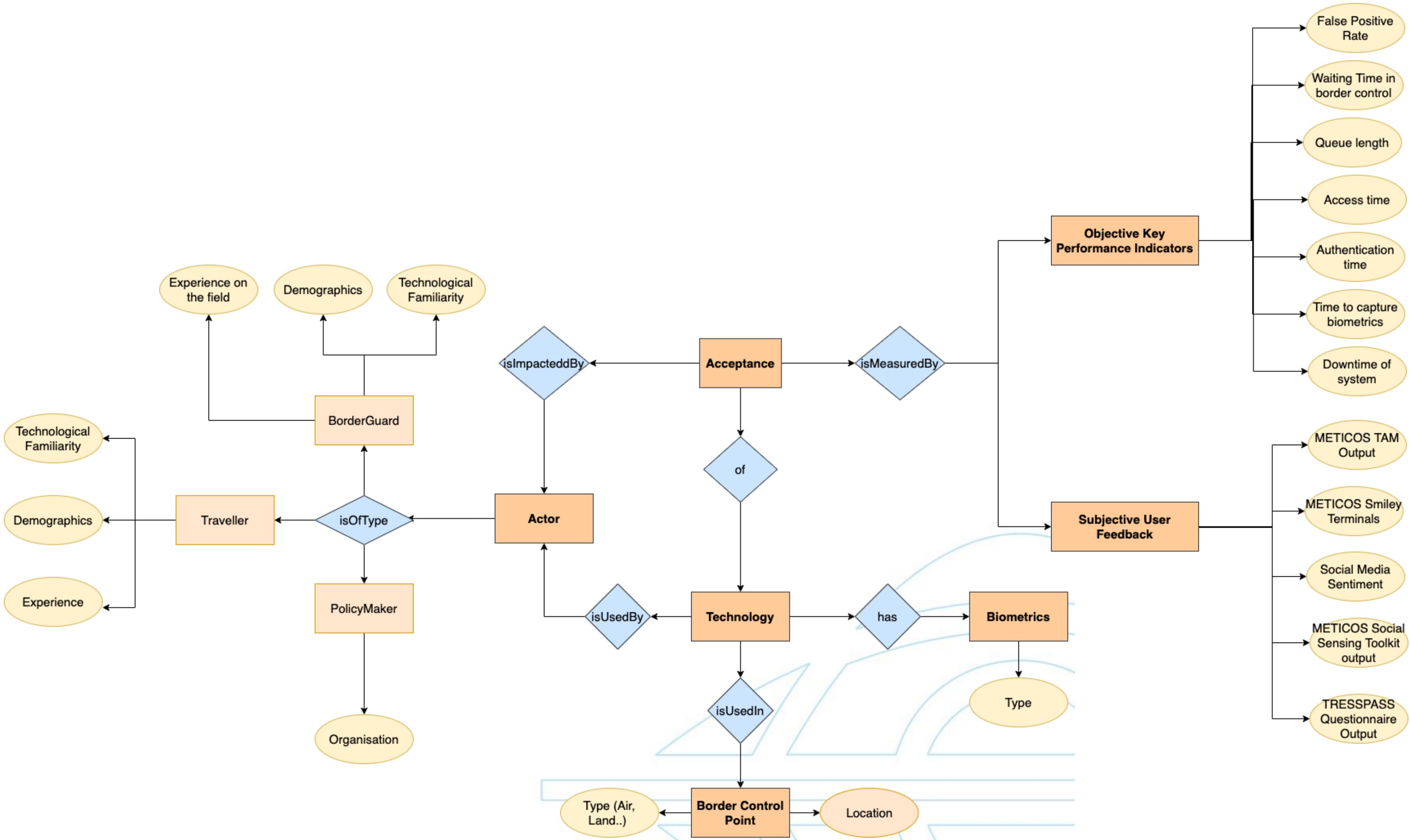

Figure 1 - METICOS Data Model

The main entities depicted and captured by METICOS Data Model are:

- Actor (this can be Border Guard, Policy Maker or Traveler)
- Acceptance
- Technology
- Border Control Point
- Biometrics
- Objective Key Performance Indicators
- Subjective User Feedback

Some of the respective attributes to the above entities are depicted in Figure 1.Please note that the rest of the attributes, metadata and relationships will be described in detail in the Deliverable 7.2.

3 Data Interoperability

All the exchanged information will be translated and presented in an understandable form, so as all METICOS components use them. CDME will contain information about the METICOS Ecosystem static information (such as actors – border guards, policy makers or travelers, procedures, equipment related to the project, installed sensors etc.) It will also include the dynamic aspects related to the METICOS Ecosystem, including queue lengths, flight delays etc., if they are made available by the BPCs. CDME will mainly interact with the METICOS middleware, which will collect all the information from the METICOS Ecosystem, translate them to a common vocabulary and store them to the CDME. So, all METICOS components will be able to store their output data to the CDME through middleware. Furthermore, CDME will be able to provide the historical data and information related to the METICOS Ecosystem to all METICOS components. This information will also be performed through middleware. In other words, each METICOS component will send a data request event to the middleware, which in sequence will communicate with CDME, retrieve the data and send them back to the component that made the request.

CDME will contain a model aiming to unify the representation of data originating from multiple heterogeneous sources and encoded in multiple diverse formats, in order to facilitate data storage and retrieval, as well as to support the interconnection and communication of diverse components of METICOS system. All the acquired information will be translated and presented in a form which is understandable by all METICOS components. Therefore, CDME relies on the defined METICOS Data Model's vocabulary, relationships, attributes and other fields in order to translate the heterogeneous data into a common language before storing them, which permits their interconnection and efficient communication.

The entity vocabulary, attributes and relationship are going to be represented using standard Hierarchical Data Formats (HDF), which are suitable for the representation of large amounts of hierarchically organized data. The HDF representations will be used as the base for the implementation in of the CDME data aggregation and storage module. By using hierarchical data models will facilitate the data minimization procedure and the scalable hierarchical analysis approach that will be followed by the analytics infrastructure for application on Big Data. The actual storage of information will be performed using a Hierarchical Data Format (HDF) adjusted to the particular data types handled within METICOS.

The access to the model is going to be realized through FIWARE Orion Context Broker [1].This will allow other components to query or update the model but also to subscribe for notifications about changes of the model.

Furthermore, CDME will ensure that all the stakeholders, by being part of a peer to peer community, will have access to up-to-dated information. Each time a new update occurs, the knowledge will be distributed to the community, coupled with additional information such as the timestamp and responsible actions or persons for the update. In this way, the history of all actions will be preserved, adding another layer of security. Within the overall community-network, several smaller peer-to-peer networks may exist, connecting entities with high correlation, thus minimizing the distribution efforts, while at the same time helping to circulate sensible data within a limited number of users.

CDME will persist the changes to the model in HDF, creating an audit trail of modifications.

The data stored as HDF can also be used as a dissemination support of the project data since HDF is a portable format, with no vendor lock-in.

CDME is also a support component for other components.

- Advance Data Mining Engine will gather the necessary data from CMDE while relying on the defined METICOS data model.
- Data that is stored in CDME will be accessed by DTVA to visualise the statistical analysis performed in the Big Data Analytics modules

4 Data storage

The data model developed for METICOS will be managed using FIWARE Orion Context Broker.

FIWARE is an open source platform for building smart solutions that are centred around a model of the world. The model consists of a set of entities and associated attributes. The model is built by gathering data from different sources (IoT, IT systems, mobile apps etc.). Then the model is used for processing, analysing and decision making. By using this approach the details about the sources of information are hidden, the application use the relevant entity attributes. Access to the model is done through a standard API known as NGSI v2. The API is using REST services and json payloads in order to query and update the model and register for changes in the model. [2] [3]

FIWARE Orion Context Broker is a FIWARE component that implements FIWARE NGSI v2 API. It stores context information updated from applications. This information consists of entities and their attributes. It enables the management of context information in a decentralized and large-scale way. By using a context broker the platform can be labelled as “Powered by FIWARE”. [3] [1]

Orion Context Broker supports operations like:

- Query context information
- Update context information
- Subscribe to notifications of context information changes
- Register context provider applications

The broker is using MongoDB for storing the context entities. MongoDB is a document-oriented database. It uses JSON-like documents with option schemas. It supports field, range query and regular-expression searches. Fields in the documents stored in the DB can be indexed with primary and secondary indices. MongoDB supports replication in order to provide high availability. It is scalable through sharding, the data is distributed across multiple shards based on a shard key. [4]

FIWARE Orion Context Broker and MongoDB were deployed on a Kubernetes cluster. For deployment we used Helm package manager for Kubernetes.

MongoDB was deployed using `bitnami/mongodb` chart available at https://bitnami.com/stack/mongodb/helm

The chart installation was done with the following command:

```
$ helm install mongodb bitnami/mongodb
```

Orion Context Broker was deployed using `fiware/orion` chart from https://fiware.github.io/helm-charts. Orion was configured to use the previously deployed MongoDB instance.

The chart installation was done with the following commands:

```
$ helm repo add fiware https://fiware.github.io/helm-charts

$ export MONGODB_ROOT_PASSWORD=$(kubectl get secret --namespace default mongodb -o jsonpath="{.data.mongodb-root-password}" | base64 --decode)

$ helm install --set 'broker.db.hosts={mongodb}' --set broker.db.auth.user=root --set broker.db.auth.password=$MONGODB_ROOT_PASSWORD orion fiware/orion
```

In Figure 2 there is a representation of corresponding Kubernetes elements created after the deployment:

```
[redacted]@meticos:~$ kubectl get deployments
NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
mongodb                  1/1     1            1           5d23h
orion                    1/1     1            1           5d20h
[redacted]@meticos:~$ kubectl get pods
NAME                                      READY   STATUS    RESTARTS        AGE
mongodb-6ddbdf5764-kc9bh                  1/1     Running   1 (4d8h ago)    5d23h
orion-6d954c5964-9mq2c                    1/1     Running   1 (4d8h ago)    5d20h
[redacted]@meticos:~$ kubectl get services
NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
mongodb                  ClusterIP   10.99.106.15    <none>        27017/TCP        5d23h
orion                    ClusterIP   10.103.10.17    <none>        1026/TCP         5d20h
```

Figure 2 Kubernetes deployment of Orion Context Broker and MongoDB

For this version of the deliverable there were no resource limits configured for the 2 components. The resource limits will be identified as the project progresses, the model will be populated with data and we will be able to estimate the data volume. Also the resources allocated will depend on the number of API calls that we will be able to estimate after integration with other METICOS components.

Another component of the CDME storage model is HDF. HDF (Hierarchical Data Forma) represents a data model, library and file format for storing and managing data. It is created to be flexible, to support efficient I/O, handle large volume of data. HDF is portable and extensible. [5] HDF is currently at version 5.

As mentioned, HDF is not a component to be deployed. On top of it we will develop a component that will implement functionalities mentions in section *3 Data Interoperability:*

- storing entity vocabulary, attributes and relationship
- storing an audit trail of model modifications

These functionalities will be implemented and presented in the next iteration of the deliverable.

5 Data access protocols and access

The authentication and authorization for CDME component will be implemented using the following FIWARE components:

- Keyrock – this component implements identity management in the FIWARE ecosystem. It allows the introduction of OAuth2-based authentication and authorisation security to services and applications. Keyrock supports the definition of users, organizations and applications. Keyrock can be configured both with a GUI interface and through an API. [6]
- Wilma – this component is used in conjunction with Keyrock in order to enforce access control to backend applications. Only certain users will be able to access CDME based on the permissions and policies to resources defined using Keyrock identity management. Wilma is a PEP Proxy (Policy Enforcement Point Proxy) that intercepts the requests to the protected application and verifies the access permissions with the identity management component (Keyrock). If the verification is successful, the request is forwarded to the destination application. [7]

In the context of the METICOS platform, these technologies will be used to allow access to CDME from the METICOS middleware. The middleware will mediate the requests to CDME from the other METICOS components.

The Keyrock component depends on a MySql database for storing the defined the users, organizations, applications, permissions and roles. The 2 components were deployed as a Helm charts.

MySql was deployed using `bitnami/mysql` chart available at https://bitnami.com/stack/mysql/helm.

The command used to install this chart is:

```
helm install mysql bitnami/mysql
```

Keyrock was deployed using the `fiware/keyrock` chart from https://fiware.github.io/helm-charts.

The commands used for deploying Keyrock are:

```
MYSQL_ROOT_PASSWORD=$(kubectl get secret --namespace default mysql -
o jsonpath="{.data.mysql-root-password}" | base64 --decode)
```

```
helm install --set db.password=$MYSQL_ROOT_PASSWORD keyrock
fiware/keyrock
```

For Wilma component we created a Helm chart based on the Docker image https://hub.docker.com/r/fiware/pep-proxy/

The chart skeleton was created with the command in folder `fiware-wilma`:

```
helm create fiware-wilma
```

After modifying it in order to integrate with the deployed Keyrock instance, Wilma was deployed using the command:

```
helm install wilma ./fiware-wilma/
```

The Kubernetes elements created after these deployments are represented in Figure 3:

```
[redacted]@meticos:~$ kubectl get all
NAME                                        READY   STATUS    RESTARTS        AGE
pod/keyrock-0                               1/1     Running   0               4d11h
pod/mysql-0                                 1/1     Running   1 (4d11h ago)   4d11h
pod/wilma-fiware-wilma-64dfdf97d7-tmgww     1/1     Running   0               4d7h

NAME                          TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/keyrock               ClusterIP   10.100.148.192   <none>        8080/TCP         4d11h
service/mysql                 ClusterIP   10.106.85.13     <none>        3306/TCP         4d11h
service/mysql-headless        ClusterIP   None             <none>        3306/TCP         4d11h
service/wilma-fiware-wilma    NodePort    10.106.49.223    <none>        1027:30507/TCP   4d8h

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/wilma-fiware-wilma    1/1     1            1           4d8h

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/wilma-fiware-wilma-64dfdf97d7   1         1         1       4d7h
NAME                          READY   AGE
statefulset.apps/keyrock      1/1     4d11h
statefulset.apps/mysql        1/1     4d11h
```

Figure 3 Kubernetes elements for Keyrock & Wilma

6 Conclusions

Deliverable D4.3 presented the CDME component of the METICOS system. At this stage of the project the main technical decisions about the component were taken. The component depends on work done on other work packages (T7.2 for the definition of data model, WP6, WP7, WP8 for the development of METICOS components that will query and update the data model), so it will evolve as the project advances. The final version of the deliverable will be presented in D4.6.

7 References

[1] "telefonicaid / fiware-orion," [Online]. Available: https://github.com/telefonicaid/fiware-orion/.

[2] "FIWARE-NGSI v2 Specification," [Online]. Available: https://fiware.github.io/specifications/ngsiv2/stable/.

[3] "CORE CONTEXT MANAGEMENT," [Online]. Available: https://fiwaretourguide.readthedocs.io/en/latest/core/introduction/.

[4] "MongoDB," [Online]. Available: https://en.wikipedia.org/wiki/MongoDB.

[5] "HDF5," [Online]. Available: https://portal.hdfgroup.org/display/HDF5/HDF5.

[6] "IDENTITY MANAGER - KEYROCK," [Online]. Available: https://fiware-idm.readthedocs.io/en/latest/.

[7] "PEP PROXY - WILMA," [Online]. Available: https://fiware-pep-proxy.readthedocs.io/en/latest/.

[8] D. Sarkar, R. Bali and T. Ghosh, Hands-On Transfer Learning with Python, Packt, 2018.

[9] S. Patel, 2 August 2020. [Online]. Available: https://suneelpatel-in.medium.com/nlp-pipeline-building-an-nlp-pipeline-step-by-step-7f0576e11d08.

[10] A. d. Juan and R. Tauler, " Chapter 8 - Data Fusion by Multivariate Curve Resolution," in *Data Handling in Science and Technology*, Elsevier, 2019, pp. 205-233.

[11] "TensorFlow," [Online]. Available: https://www.tensorflow.org/. [Accessed 2 7 2021].

[12] "SpaCy," [Online]. Available: https://spacy.io/. [Accessed 2 7 2021].

[13] "SciPy," [Online]. Available: https://www.scipy.org/. [Accessed 2 7 2021].

[14] "Scikit-learn," [Online]. Available: https://scikit-learn.org/. [Accessed 2 7 2021].

[15] "polyglot," [Online]. Available: https://polyglot.readthedocs.io/en/latest/. [Accessed 2 7 2021].

[16] "Grafana Overview," [Online]. Available: https://grafana.com/grafana/. [Accessed 1 7 2021].

[17] "Gensim," [Online]. Available: https://radimrehurek.com/gensim/. [Accessed 2 7 2021].

[18] "CoreNLP," [Online]. Available: https://stanfordnlp.github.io/CoreNLP/. [Accessed 2 7 2021].

[19] "Best Python Libraries for Data Science," 30 05 2021. [Online]. Available: https://analyticsindiamag.com/best-python-libraries-for-data-science-in-2021/.

[20] "Analytics Vidhya," 1 5 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/top-8-python-libraries-for-natural-language-processing-nlp-in-2021/.