Common European Data Spaces and Robust AI for Transparent Public Governance

CEDAR

Project acronym: CEDAR

Project full title: Common European Data Spaces and Robust AI for Transparent Public Governance

Call identifier: HORIZON-CL4-2023-DATA-01

Type of action: HORIZON-RIA

Start date: 01/01/2024

End date: 31/12/2026

Grant agreement no: 101135577

D3.1 DataOps, MLOps, and Secure CEDS Connectors

Document description: Results from WP3 tasks, initially focusing on the SotA analysis and refining research gaps, and then presenting results addressing them.

Work package: WP3

Author(s): Nicola Leonardi (ENG), Davide Profeta (ENG)

Editor(s): Davide Profeta (ENG)

Leading partner: ENG (Engineering Ingegneria Informatica S.p.A)

Participating partner: ART, CEA, CERTH, ICS, INS, NCI, SNEP, TRE, UBI, UPM, VICOM

Version: 1.0 Status:Submitted

Deliverable type: Report Dissemination level: PU

Official submission date: 30/06/2024 Actual submission date: 28/06/2024

The CEDAR project has received funding from the European Union's Horizon Europe project call HORIZON-CL4-2023-DATA-01 funded project Grant Agreement no. 101135577

Disclaimer

The CEDAR consortium consists of the following partners:

No.	Partner Organization Name	Partner Organization Short Name	Country
1	Centre for Research and Technology Hellas	CERTH	Greece
2	Commissariat al Energie Atomique et aux Energies Alternatives	CEA	France
3	CENTAI Institute S.p.A.	CNT	Italy
4	Fundacion Centro de Technologias de Interaccion Visual y Comunicaciones VICOMTECH	VICOM	Spain
5	TREBE Language Technologies S.L.	TRE	Spain
6	Brandenburgisches Institut für Gesellschaft und Sicherheit GmbH	BIGS	Germany
7	Christian-Albrechts University Kiel	KIEL	Germany
8	INSIEL Informatica per il Sistema degli Enti Locali S.p.A.	INS	Italy
9	SNEP d.o.o	SNEP	Slovenia
10	YouControl LTD	YC	Ukraine
11	Artellence	ART	Ukraine
12	Institute for Corporative Security Studies, Ljubljana	ICS	Slovenia
13	Engineering – Ingegneria Informatica S.p.A.	ENG	Italy
14	Universidad Politécnica de Madrid	UPM	Spain
15	Ubitech LTD	UBI	Cyprus
16	Netcompany-Intrasoft S.A.	NCI	Luxembourg
17	Regione Autonoma Friuli Venezia Giulia	FVG	Italy
18	ANCEFVG – Associazione Nazionale Costruttori Edili FVG	ANCE	Italy
19	Ministry of Interior of the Republic of Slovenia / Slovenian Police	MNZ	Slovenia
20	Ministry of Health / Office for Control, Quality, and Investments in Healthcare of the Republic of Slovenia	MZ	Slovenia
21	Ministry of Digital Transformation of the Republic of Slovenia	MDP	Slovenia
22	Celje General Hospital	SBC	Slovenia
23	State Agency for Reconstruction and Development of Infrastructure of Ukraine	ARU	Ukraine
24	Transparency International Deutschland e.V.	TI-D	Germany
25	Katholieke Universiteit Leuven	KUL	Belgium
26	Arthur's Legal B.V.	ALBV	Netherlands
27	DBC Diadikasia	DBC	Greece
28	The Lisbon Council for Economic Competitiveness and Social Renewal asbl	LC	Belgium
29	SK Security LLC	SKS	Ukraine
30	Fund for Research of War, Conflicts, Support of Society and Security Development – Safe Ukraine 2030	SU	Ukraine
31	ARPA Agenzia Regionale per la Protezione dell'Ambiente del Friuli Venezia Giulia	ARPA	Italy

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.1	20/05/2024	ToC released; partners matched to sections	Silvio Sorace (ENG)
0.2	05/06/2024	Contribution to Sections 2 and 3	Nicola Leonardi (ENG)
0.3	05/06/2024	Contribution to Section 4	Davide Profeta (ENG)
0.4	06/06/2024	Contribution to Section 5	Anastasios Nikolakopoulos (NCI)
0.5	10/06/2024	Contribution to Section 6	Jolanda Modic (ICS)
0.6	21/06/2024	Internal review	Giulia Preti (CNT)
0.7	24/06/2024	Internal review	Victor Drobotenko (SKS)
1.0	28/06/2024	Final version addressing all comments	Silvio Sorace (ENG)

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise.

Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Table of Contents

List of Figures

List of Terms and Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
BoW	Bag of Words
BDA	Big Data Analytics
CD	Continuous Delivery
CI	Continuous Integration
CM	Continuous Monitoring
CT	Continuous Training
DoA	Description of Action
DVC	Data Version Control
CEDS	Common European Data Spaces
CRISP-DM	Cross-Industry Standard Process for Data Mining
DevOps	Software Development and IT Operations
DSML	Data Science and Machine Learning
EDA	Exploratory Data Analysis
ETL	Extract, Transform, Load
ETSI	European Telecommunications Standardization Institute
EU	European Union
GNN	Graph Neural Network
IDSA	International Data Spaces Association
KG	Knowledge Graph
ML	Machine Learning
MVG	Minimum Viable Graph
NGSI-LD	Next Generation Service Interfaces – Linked Data
nIDS	network Intrusion Detection System
NLP	Natural Language Process
PCA	Principal Component Analysis
RFE	Recursive Feature Elimination
SDM	Smart Data Models
SFS	Sequential Feature Selection
SotA	State of the Art
TF-IDF	Term Frequency-Inverse Document Frequency
WP	Work Package

Executive Summary

WP3 provides innovative solutions for data collection and cleaning. Modern Data and ML Ops (operations) technologies will improve quality, reliability, and scalability of data processing. The integration of the new data sources with CEDS will generate social and economic benefits by increasing the availability of data (in volume and quality) for socio-economic applications across the EU. WP3 is also dedicated to the development (and penetration testing) of data management technologies following all the security requirements and specifications defined in WP1.

1 Introduction

Within the CEDAR project, WP3 is responsible, in alignment with the roadmap defined in WP1 and data modelled in WP2, for the efficient, scalable, secure management of big data and their integration with CEDS. It also ensures cybersecurity of CEDAR technologies through cybersecurity risk assessment, security mechanisms, and penetration testing. To facilitate these goals, DataOps, MLOps, and CEDS connectors will be delivered.

1.1 Positioning of the Deliverable within CEDAR

D3.1 represents the first of three documents. It focuses on the SotA analysis and refining research gaps, while the next two versions (D3.2 and D3.3) will describe the developed technologies.

1.2 Structure of the Deliverable

The information in this document is structured as follows:

- Section 1 provides a brief overview of the work package objectives
- Section 2 provides an overview of the CRISP-DM model for directing data mining projects, but it can also be applied to generic machine learning projects
- Section 3 describes the DataOps pipeline to improve collaboration and productivity among data scientists, data engineers, and other data professionals
- Section 4 presents the MLOps methodology for the efficient management, deployment and monitoring of ML models
- Section 5 presents the CEDS and their integration to develop dataspace connectors that align with key European initiatives (like IDSA, GAIA-X, EOSC, etc.) to create a secure, reliable, and integrated European data network
- Section 6 describes how to conduct a (manual) continuous cybersecurity risk assessment of the CEDAR assets, and accordingly proposes technical and organisational measures that will drive the design, development, deployment, and use of the CEDAR solutions
- Section 7 provides conclusions and future work is briefly addressed

2 The CRISP-DM model

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is a well-established method for directing data mining projects.

Published in 1999 to standardize data mining processes across various industries, it has since become the most prevalent methodology for data mining, analytics, and data science projects. Even today, CRISP-DM remains the most widely used approach for data science projects [1].

There are two main ways of interpreting the method:

- As a *Methodology* and guide, it outlines the typical phases of a project, details the tasks associated with each phase, and explains the relationships and dependencies between these tasks. Essentially, it offers a structured approach to planning a data mining project, addressing the question of "How to do it".
- As *a Process Reference* model, CRISP-DM gives an overview of the data mining life cycle, describing common approaches used by data mining experts. It answers the question of "What to do".

The main features of CRISP-DM can be summarized in the following:

- Non-proprietary
- Application/Industry neutral
- Tool and technique independent
- Support documentation of projects
- Focus on business issues / As well as technical analysis
- Framework for guidance
- Experience-based / Templates for Analysis
- Support knowledge transfer and training

Although the CRISP-DM model was designed for data mining, it can also be applied to generic machine learning projects. This method can be used for project planning and management, for communicating, and for documentation purposes.

The best way to understand this methodology is to arise the question “Why Should There be a Standard Process”?

There are several answers, but the main ones are:

- The data mining process must be reliable and repeatable by people with little data mining background.
- It is good having a framework for recording experience, indeed it allows projects to be replicated.
- It represents aid to project planning and management.
- It is a “comfort factor” for new adopters.
- It encourages best practices and helps to obtain better results.

The entire CRISP-DM lifecycle can be described by the following diagram.

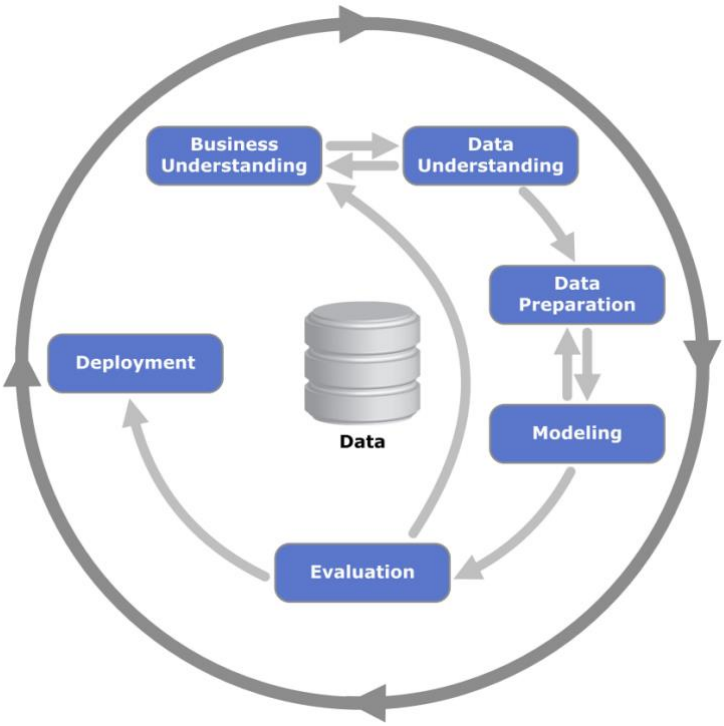

Figure 1. CRISP-DM Life Cycle [2]

The life cycle model consists of six phases with arrows illustrating the most significant and frequent dependencies between them. These phases can be seen as a set of guardrails to help the planning, organization and implementation of a data mining project. The sequence of the phases is not strict. In fact, most projects move back and forth between phases as necessary. It depends on the outcome of each phase which phase or which task of a phase, has to be performed next. The outer circle in Figure 1 symbolizes the cyclical nature of data mining itself. A data mining project is not over once a solution is deployed. After deployment, it is necessary to monitor the application and tune it in the so-called Continuous Training CT and Continuous Monitoring CM steps.

The six phases are:

- Business Understanding: project objectives and requirements understanding, data mining problem definition.
- Data Understanding: initial data collection and familiarization, data quality problems identification.
- Data Preparation: table, record and attribute selection, data transformation and cleaning.
- Modeling: modeling techniques selection and application, parameters calibration.
- Evaluation: business objectives & issues achievement evaluation.
- Deployment: model deployment, repeatable data mining process implementation.

The CRISP-DM model is adaptable and can be easily customized. Each project may have characteristics that push the effort on some of these specific phases (e.g. focus on data exploration and visualization more than deployment phases) but it is still good to take into consideration all the phases and the issues they could arise if the project aims for a long-term solution and coherent data mining goals.

The CRISP-DM can be seen as a waterfall approach and an agile approach based on different points of views [1].

The reporting requirements for each step bring the methodology to the waterfall area. There is also a note in the business understanding phase that states "the project plan contains detailed plans for each phase. For example, decide at this point which evaluation strategy will be used in the evaluation phase".

On the other hand, CRISP-DM indirectly promotes agile principles and practices by stating: "The sequence of the phases is not rigid. Moving back and forth between different phases is always required. It depends on the outcome of each phase which phase or which particular task of a phase, has to be performed next. The arrows indicate the most important and frequent dependencies between phases".

Therefore CRISP-DM allows flexibility in the implementation of the overall project by allowing choice between the Horizontal Slicing way typical of the waterfall approach and the Vertical Slicing typical of the agile approach [1]. Both methodologies are fine, and the final choice depends on the final use case that can take into consideration several other aspects of the project such as the number of partners that work on the activities, the hardware/software availability, etc.

2.1 The six phases of CRISP-DM

The following figure presents an outline of the 6 phases, each of which accompanied by tasks (bold) and required outputs (italic).

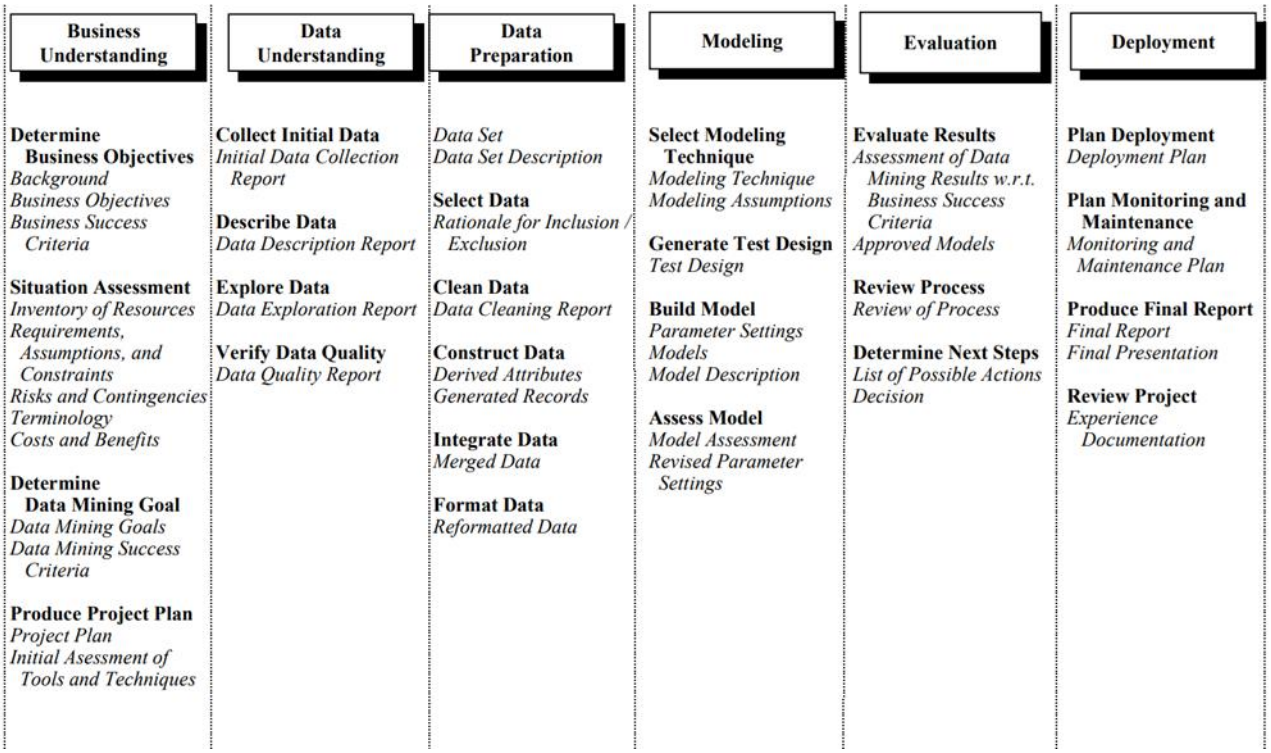

Figure 2. The 24 tasks with the corresponding outputs of the CRISP-DM model [2]

The next paragraphs will describe each phase in detail.

2.1.1 Business Understanding

The Business Understanding phase is focused on good problem definition and ensuring that the business's problem is solved. The final goal is to focus on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

The four tasks in this phase are:

- Determine business objectives
- Assess situation
- Determine data mining goals
- Produce project plan

Determine Business Objectives

The data analyst's primary objective is to thoroughly understand the client's goals from a business perspective. The aim is to identify key factors early on that could impact the project's outcome. A possible consequence of neglecting this step is to expend a great deal of effort producing the right answers to the wrong questions.

Assess Situation

This task involves a more detailed investigation into all resources, constraints, assumptions, and other relevant factors. While the previous step is more general, this step requires enriching the details.

In addition to the *Inventory of Resources* (data, software, computing power) one of the expected relevant outputs are the lists of *Requirements, Assumptions, and Constraints* (Figure 2) that should contain information about any data security concerns as well as any legal issues, and also consideration about the legitimacy of the use of the data.

Determine Data Mining Goals

While a business goal states objectives in business terminology, a data mining goal states project objectives in technical terms. The focus of the task is exactly creating this mapping, defining the data mining problem type (e.g., classification, prediction, clustering), specifying the criteria for model assessment and for assessment (in terms of metrics e.g., accuracy) of the predictive task.

Produce Project Plan

This task will outline the intended plan for achieving the data mining objectives and, consequently, the business goals. The plan should specify the anticipated set of steps to be performed during the rest of the project including an initial selection of tools and techniques.

One of the required outputs is the *Initial Assessment of Tools and Techniques* (Figure 2). After the initial phase, the project conducts an initial assessment of tools and techniques. This involves selecting a data mining tool that accommodates various methods for the different stages of the process. Assessing tools and techniques early in the process is crucial as their selection can significantly impact the entire project.

2.1.2 Data Understanding

Data collection and understanding is the second step in the CRISP-DM framework. In this step, there is a deeper dive to understand and analyze the data for the problem statement formalized in the previous step. This step begins with investigating the various sources of data outlined in the detailed project plan. These sources of data are then used to collect data, analyze different attributes, and make a note of data quality. This step also involves what is generally termed exploratory data analysis (EDA).

The tasks in this phase are:

- Collect initial data
- Describe data
- Explore data
- Verify data quality

Collect Initial Data

The task is on the data acquisition effort (or access to the data) listed in the project resources; it includes data loading if necessary for a deep data flow understanding. If there are multiple data sources, integration and harmonization are additional issues, either here or in the later data preparation phase.

The output, the *Initial Data Collection Report,* should contain a clear description of the location of the datasets, the methods for collecting the data, the problems encountered, and the solutions implemented.

Describe Data

The focus is on examining the "gross" or "surface" properties of the acquired data and reporting on the results. Its output, the *Data Description Report*, should describe the data that was acquired, including their format, their structures/attributes (e.g. categorical, ordinal, numerical, discrete, continuous), and the amount of data (e.g. the number of records and fields in each table).

Explore Data

This task addresses data mining questions, which can be explored using querying, visualization, and reporting techniques. It may involve analyzing feature distributions and relationships between attributes. This step represents what is generally termed exploratory data analysis (EDA) and lays down the foundation for the next phase and hence it cannot be neglected at all.

The output, the *Data Exploration Report*, should describe the analysis in terms of statistical indicators, distribution of key attributes, results of simple aggregation, but also provide charts and plots that summarize the characteristics of the data and lead to interesting insights for a further examination.

Verify Data Quality

The focus here is the assessment of the quality of the data, addressing questions such as completeness and correctness.

The entire project has to face the fact that low data quality makes it impossible to trust the results of any analysis: the "garbage in, garbage out" principle.

Regarding data quality, it is possible to go into more detail descriptors such as:

- Syntactic accuracy: the entry is not in the domain. Examples: spelling error on text, words in numerical attributes
- Semantic accuracy: the entry is in the domain but not correct. Example: John Smith is female. It needs more information to be checked.
- Unbalanced data: the dataset might be extremely biased to one type of record.
- Timeliness: it is possible that the data is not updated correctly.
- Outliers: the presence of outliers has to be understood.
- Missing values types: missing values can be classified as missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR) [3].
- Duplicated values: try to understand if there are duplicate values and the reasons for this.

The *Data Quality Report* output should document the results of data quality verification. If any quality issues are identified, potential solutions should be listed alongside them.

2.1.3 Data Preparation

This is the third and the most time-consuming step in any data science project. Data preparation takes place when there is a complete understanding of the business problem and data availability is achieved. This step involves data integration, cleaning, wrangling, feature selection, and feature engineering. Data preparation is the most time-consuming step, taking over 70% [63], [64] of the overall time taken for any data science project. It covers all activities to construct the final dataset from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order.

The specific tasks are:

- Select data
- Clean data
- Construct data
- Integrate data
- Format data

Select Data

This task revolves around determining the data to be utilized for the analysis. Criteria considered include relevance to the data mining goals, quality and technical constraints such as limitations on data volume or types. This process includes common procedures like feature selection and record selection.

Clean Data

The goal is to elevate the data quality to meet the standards required by the selected analysis techniques. This could entail selecting clean subsets of the data and implementing appropriate defaults.

This step has to handle:

- Anomalous values: missing values, unknown values (not meaningful values), not valid values (not significant values).
- Outliers: data points that seem to be generated by a different process than the other data.
- Duplicate entries: entries that are duplicated for some reason and could affect the final evaluation of the model.
- Bad values: values that are not fresh or do not reach an acceptable level of quality.

For each of these general cases there is the need to find a way to manage them (e.g. elimination, imputation). As a side note, sometimes outlier analysis is the ultimate goal of the project, and thus, elimination and imputation are not good approaches in this specific use case.

Its output, the *Data Cleaning Report,* should describe what decisions and actions/transformations were taken to address the data issues considering the possible impact on the final analysis results. Everything that arose during the Data Understanding Phase should be fixed and managed here.

Construct Data

This task includes constructive data preparation operations such as:

- Features engineering: the production of new derived attributes, e.g. one-hot encoding, datetime encoding.
- Data augmentation: the construction of new records.
- Data transformation: the transformation of values for existing attributes due to, for example, strong asymmetry in the data and presence of peaks. Common methodologies are min-max normalization, z-score normalization and logarithmic transformation.

- Features extraction: e.g. Principal Component Analysis (PCA), Autoencoders, Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and embedding, in the case of text data and image data.

Its outputs should define each new attribute that derives from an existing one and, for the newly generated records, a description of the underlying procedure should be documented. In the case of synthetic data generation, it is useful to quantify the similarity with the real data (e.g. KS-tests validation metric).

Integrate Data

This task involves combining data from multiple sources, tables, or records to generate new records or values. It may also encompass activities aimed at enriching data by merging information from different data sources.

Format Data

Formatting transformations primarily involve syntactic adjustments to the data, which maintain its meaning but may be necessary for compatibility with the modeling tool. This can encompass actions such as converting text to lowercase, trimming entries, or other adjustments to meet the specific requirements of the modeling tool (e.g., data order, attribute arrangement).

Data harmonization techniques can be considered as part of this kind of data manipulation procedure.

2.1.4 Modeling

Once the data has been transformed through the previous steps, it is time to define the appropriate algorithms, settings, and hyperparameters. Various machine learning techniques can be analyzed, individually or in combination, tailored to the specific dataset and project needs uncovered in the previous phases. These methods encompass supervised methods like classification or regression, unsupervised methods like clustering, and potentially hybrid approaches merging different techniques (ensemble models).

The tasks that compose this phase are:

- Select Modeling Technique
- Generate Test Design
- Build Model
- Assess Model

Select Modeling Technique

The focus is the selection of the actual modeling technique that is to be used. This technique has to be described in a specific and technical way. If multiple techniques are used, each of that has to be described accordingly.

The required outputs are:

- Modeling techniques: a document that describes the algorithm/s used (e.g. decision tree, neural network)
- Modeling assumptions: Identify any built-in assumptions made by the technique about the data (e.g. quality, format, distribution). It is also necessary to compare these assumptions with those in the *Data Description Report* and make sure that these assumptions hold.

Generate Test Design

Before constructing a model, it is essential to devise a procedure or mechanism to assess the model's quality and validity. A common practice involves dividing the dataset into training, validation, and testing sets, and determining the appropriate metrics (or set of metrics, such as optimizing and satisficing metrics) to evaluate the model.

Build Model

Execute the modeling tool on the prepared dataset to generate one or multiple effective models. These are the ready-to-use versions of the models (e.g. serialized version, weights) not a simple report.

There is also the need to produce the description of the model in terms of set of hyper-parameters, training procedures (e.g. learning rate scheduling) and description of behaviors, limitations, and ways of interpreting its outputs.

Assess Model

The models are evaluated technically according to the metrics specifically chosen for the algorithm's purpose. The difference between this task and the following *Evaluation Phase* is that this task only considers models, whereas the evaluation phase also takes into account all other results produced in the course of the project.

The *Model Assessment* output summarizes the results by outlining the attributes of the generated models, such as accuracy, and ranking their quality in comparison to each other. Seek feedback from domain experts to validate the plausibility of the results and the modeling approach.

As mentioned at the beginning of the description of the CRISP-DM methodology, based on these results, it is often necessary to go back to the data preparation phase if the desired goals are not reached.

2.1.5 Evaluation

Model evaluation is the process of evaluating the built model against certain criteria to assess its performance. Model performance is usually numerical values that help to extract the effectiveness of the model. These are the so-called optimizing metrics, but, for the success of the entire project, other types of metrics have to be taken into account such as the satisfactory metrics (e.g. throughput, running time, resource allocation) and the degree to which the model meets the business objectives.

This phase can be split into:

- Evaluate Results
- Review Process
- Determine next steps

Evaluate Results

While the previous step is purely technical, this step focuses on evaluating how well the model aligns with the business objectives and aims to identify any potential deficiencies from a business standpoint.

Its output is a report that condenses the assessment findings in relation to the criteria for business success, concluding with a statement regarding whether the project currently fulfills the initial business objectives.

Review Process

This step overviews the data mining process highlighting possible overlooked factors or tasks (e.g. verify that the model in prediction uses only the data already available and does not "go ahead" looking future data). It is the step where possible missed activities or activities that could have been implemented in a different way can be reported.

Determine Next Steps

Based on the evaluation outcomes and process review, the project determines its course of action at this juncture. It must decide whether to conclude the project and proceed to deployment, if deemed suitable or to initiate additional iterations or new data mining projects. This task involves analyzing the remaining resources and budget, which play a significant role in shaping these decisions. It is also possible to recommend alternative continuations.

The list of possible actions and the selection of at least one of these is the final output of this phase.

2.1.6 Deployment

The last phase is related to the final definition of a strategy for making the data mining result available to the stakeholders. It focuses on questions like how the results need to be utilized, who needs to use them, and how often do they need to be used.

The knowledge gained will need to be organized and presented in a way that the customer can use it. However, depending on the requirements, the deployment phase can be as simple as generating a report (a dissemination effort) or as complex as releasing a complete infrastructure that delivers a service to end users.

The phase can be decomposed into:

- Plan deployment
- Plan monitoring and maintenance
- Produce final Report
- Review Project

Plan Deployment

To deploy the data mining results into the business, use the evaluation results to develop a deployment strategy. This may also involve documenting the procedure for future deployment and identifying potential issues that could arise during the deployment of the data mining results.

Plan Monitoring and Maintenance

Effective monitoring and maintenance are crucial considerations when integrating the data mining results into daily business operations and its ecosystem. Strategically preparing a maintenance plan is essential to prevent prolonged periods of incorrect usage of data mining results. To oversee the deployment of the data mining results, the project requires a comprehensive monitoring plan tailored to the specific deployment type. This plan meticulously considers the specificity of the deployment scenario. It is also better if it takes into consideration the changes over time (the dynamic aspect of the system).

Produce Final Report

At the end of the project, the project leader and their team compile a final report. This report may vary depending on the deployment plan. It could serve as a summary of the project and its experiences or it may constitute a comprehensive presentation of the data mining results, providing a thorough overview of the outcomes and insights gained. It is worth to mention that different target audiences could require different types of reports.

Review Project

This stageassesses how well the initial data mining goals have been met, describing what went right and what failed and possible improvement. It could also include interviews with all the relevant people involved in the project. It is important to abstract from details to make the experience useful for future projects.

The output of this stage, the *Experience documentation,* should condense the key experiences gained throughout the project, encompassing insights into pitfalls encountered, misleading approaches, or guidance for selecting the most appropriate data mining techniques in similar contexts.

2.2 CRISP-DM and CEDAR project

The approach to adopting the CRISP-DM methodology is a project choice and can have different nuances. Some projects require a precise and timely adoption including all the required output documentation, while other projects may consider the methodology as a checklist to follow.

Given the nature of the CEDAR project as a European project with clear and predefined improving steps and formal deliverables, a not too strict adherence is considered more appropriate, trying to assimilate the best from the framework without too much stress on the activities. From this perspective, the following list summarizes the 10 principles considered most important for carrying out the CRISP-DM process.

1. Understand the Project Goals: ensure a clear understanding of the objectives and requirements of the project. TProper criteria should also be described to determine a successful or useful outcome.
2. Situation Assessment: includes resource inventory (the list of resources available for the project including ICT needs, software, platforms, etc.) and any data security concerns as well as legal and privacy data issues.
3. Data Documentation: review any available documentation for the datasets, including data dictionaries, metadata, or other documentation that provides information about how the data was collected, processed, and any known limitations or biases. The required data processing mode (online synchronous-asynchronous, batch) should be described to drive the DataOps architecture development.
4. Data Collection: this includes the description of dataset locations, dataset format, methods for collection (e.g. API, SQL) and possible problems (limitation of use, etc.). Thedescription alone is not enough, and the acquisition effort is required. The data from the sources listed in the project resources have to be downloaded (at least a representative portion) for a deep data understanding.
5. Data Exploration: exploring the datasets to understand the underlying patterns, relationships, and trends. Statistical summaries, charts, and graphs can help to gain insights into what each variable represents and how it relates to the problem at hand (EDA analysis).
6. Data Profiling: conduct data profiling to summarize the characteristics of the data, such as distributions, missing values, outliers, and correlations. This can help identify anomalies or issues that may affect the relevance of the data.
   Data Profiling also covers the Data Quality Assessment with focus on data accuracy, completeness, consistency, and timeliness. Low-quality data may not be relevant or may require preprocessing to improve its relevance.
7. Modeling Technique: describe the type of problem to address (e.g. classification, regression, clustering) and report some State of the Art (SotA) algorithm/s available in that context (e.g. decision tree, neural network).
8. Generate Test Design: before building a model there is the need to generate a procedure or mechanism to test the model's quality and validity by describing the intended plan for training, testing, and evaluating the models, defining what metrics to use to evaluate the output results, and finally splitting the entire dataset into train/val/test set. This step is important to link the data to the models because in the project there is the need to build ML models that requires labeled dataset and clear predefined metrics to be evaluated on.
9. Data Relevance to the Problem: assess how each data source (and each variable in the dataset) relates to the specific problem or question being addressed by the project. Focus on data sources (variables) that are directly relevant to the problem and may have predictive or explanatory power. Refer to documented use cases and literature reports to get feedback about the correct use of the data source type with the task at hand.
10. Consultation and Domain Knowledge: Discuss the relevance of the data with stakeholders, including domain experts or subject matter. Their input can provide valuable insights and help ensure that the data

being used aligns with the project goals and requirements. Understanding the domain can boost and streamline the overall process.

2.3 CRISP-DM Benefits and limitations

There are several reasons why the CRISP-DM model is still widely used even more than 25 years after its introduction. Some of these are the following:

- it provides a uniform framework for both guidelines and experience documentation;
- it is flexible, as it accounts for differences in both data types and business problems;
- it is aligned with the requirement that a data mining process must be reliable and repeatable by people with little data mining skills;
- placing data at the center is a de facto Data-Centric approach; indeed, there is a focus on data at each phase of the process. "CRISP-DM provides strong guidance for even the most advanced of today's data science activities" (Vorhies, 2016) [65].

Some shortcomings of the process are related to the fact that it is "not modern" regarding, for example, Data Security and Data Privacy aspects (although something related is mentioned in the task "Assess Situation" in the Business Understanding phase).

Data Security concerns safeguarding information against unauthorized access or malicious attacks, while Data Privacy prioritizes the protection of users' and groups' rights regarding their own data. Data security typically encompasses aspects such as data confidentiality, access control, infrastructure security, and system monitoring, employing technologies like encryption, trusted execution environments, and monitoring tools. In contrast, data privacy focuses on privacy policies and regulations, data retention and deletion policies, data-subject access requirement (DSAR) policies, management of data usage by third parties, and obtaining user consent.

Another limitation is the lack of consideration regarding the Data Ethics topic. Impact on individuals, organizations, and society, ethical and normative concerns, bias in data and algorithmic and regulatory issues are concepts that do not have place in the process framework. Furthermore, there is no connection with the FAIR principles (Findability, Accessibility, Interoperability, and Reusability) that a process must satisfy to be aligned with current best practices in a Data Science project.

When CRISP-DM was released, the deployment phase often ended with a compilation and dissemination of analysis results and their explanation.

Nowadays the goal is more ambitious, and the production of a real working service is now the standard. In this setting arises the MLOps (Machine Learning Operations) framework that engages four pillar concepts:

- Continuous Integration (CI): testing and validating code, components, data, data schemas and models.
- Continuous Delivery (CD): not only about deploying a single software package or a service, but a system which automatically deploys another service.
- Continuous Training (CT): the process, specific to ML systems, that automatically retrains candidate models for testing and serving.
- Continuous Monitoring (CM): catching errors in production systems and monitoring production inference data and model performance metrics (e.g. monitoring data drift, concept drift).

These aspects are only partially covered by the Plan Monitoring and Maintenance task in the deployment phase, while are very important aspect to take into account from the very first step of the process (inside business understanding phase) because of all the implications that could arise.

To summarize, an updated version of CRISP-DM could include tasks related to Data Security, Data Privacy, Data Ethics, and specific MLOps activities.

In the Cedar project, all these aspects will be taken into consideration from the very first stages of analysis and prototyping, effectively constituting an enhanced version of the classic CRISP-DM model.

2.4 CRISP-DM and DataSpaces

Often, an obstacle to a wider deployment of an effective Data Science project is the lack of trust in data sources and sharing mechanisms. To counter this problem, the International Data Space Association (IDSA) has developed an architecture to define a standard for data exchange on a trusted and self-regulated basis.

IDSA defines a reference architecture, which supports sovereign exchange [4] and sharing of data between partners independent from their size and financial power, where data sovereignty can be defined as "a natural person's or corporate entity's capability of being entirely self-determined with regard to its data" [5]. In practical terms, it empowers companies and individuals to decide for themselves how, when, and at what price their data is utilized throughout the value chain, thereby facilitating new intelligent services and innovative business processes.

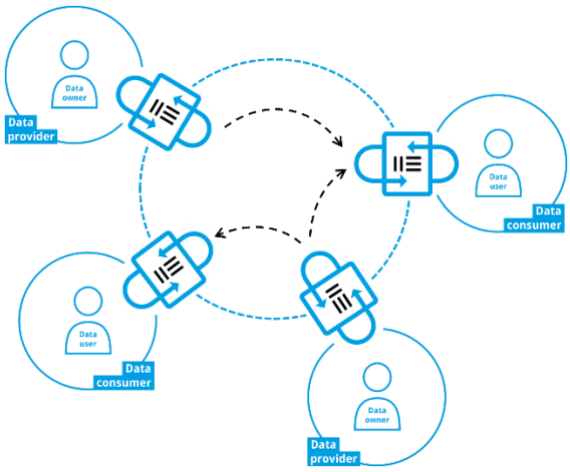

Figure 3. Data sovereignty concept in [5]

So, the sovereignty framework ensures that data is stored and processed in compliance with the laws and regulations of the country where the data originates. This is achieved using data controllers, who are responsible for ensuring that data is processed in accordance with local laws and regulations.

Another key aspect of the IDS standards is the implementation of data protection and privacy by design principles. This ensures that privacy and security are integrated into the data-sharing process from the beginning, rather than being added later. This is accomplished through techniques like data anonymization and pseudonymization, along with the use of secure communication protocols.

The IDS standards also offer guidelines for data interoperability, crucial for effective data sharing and collaboration. This involves using common data formats and protocols, as well as employing semantic data models to ensure data is easily understood and utilized by different organizations [5].

Additionally, the IDS standards provide guidelines for data governance, which are vital for responsible and ethical data use. This encompasses data use agreements and data access policies, as well as data quality and data lineage techniques to ensure data accuracy and reliability.

The CEDAR project, aiming at the adoption of IDS standards, will cover the shortcomings of the classic CRISP-DM methodology, aligning itself with the requests and complexity of a modern data science project.

3 The DataOps Pipeline

DataOps, similar to DevOps, is a methodology that combines data engineering, data integration, and data quality practices with the aim of improving collaboration and productivity among data scientists, data engineers, and other data professionals. The steps in a DataOps pipeline can vary depending on the specific requirements of a project or organization, but generally, it follows a similar structure:

- Data Collection: also called data ingestion, involves collecting data from various sources such as databases, APIs, files, streaming platforms, etc.
- Data Cleaning and selection: this step involves tasks like data cleaning, data deduplication, and feature selection.
- Data Construction: it could include feature normalization, feature engineering, feature extraction, and data harmonization.
- Data Formatting: additional syntactic transformation to better suit the final goal (e.g. from a standard table to a graph representation).
- Data Storing: the processed data must be stored in a suitable data storage system for further analysis. Common storage options include data lakes, data warehouses, NoSQL databases, vector DB, graphDB or traditional relational databases.

3.1 DataOps and CEDAR project

From the feedback from the pilots, it emerged that within the CEDAR project different types of data will have to be managed such as text, images, and tabular data. Furthermore, the ultimate goal is to have a knowledge graph as final output, capable of enclosing all the possible relationships between the entities described in the data to gain a strong semantic representation suitable for state-of-the-art graph neural network modeling techniques.

The following figure summarizes the idea through the use of interconnected blocks. The pipeline concept is often associated with the concept of DAG (Direct Acyclic Graph), a conventional approach used by several tools to implement standard data pipelines.

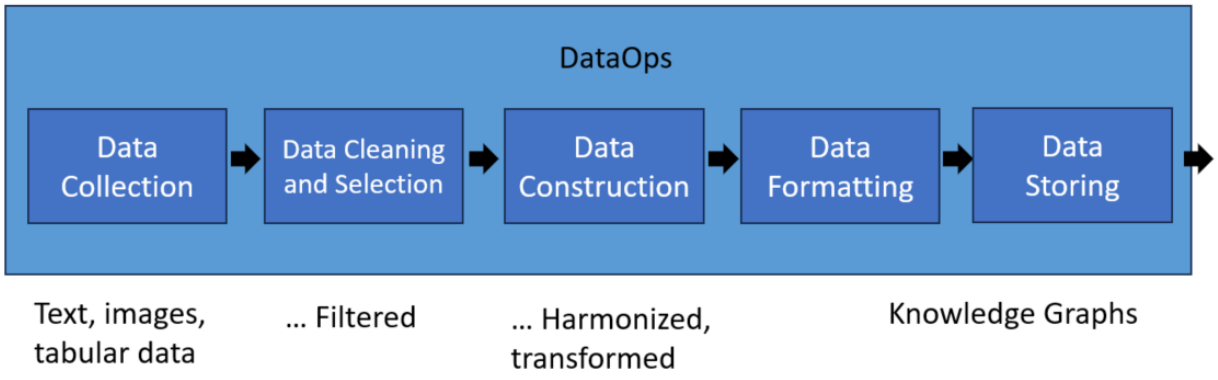

Figure 4. The standard DataOps Pipeline

The next paragraphs will analyze in detail the three families of data exposed above in terms of standard steps within the pipeline. Furthermore, particular attention will be given to initial data quality control to apply the correct data cleaning techniques.

3.2 Textual Data

Structured data types, encompassing both numerical and categorical features, are more straightforward to handle and integrate into machine learning workflows. Textual data stands out as a significant source of unstructured

information, posing various challenges such as syntactical comprehension, semantic interpretation, format discrepancies, and content intricacies. Moreover, textual data necessitates meticulous preprocessing and cleaning steps to convert it into a numerical format usable by machine learning algorithms, highlighting the importance of rigorous feature engineering.

Some useful techniques to cover the data-quality assessment in the context of text domain are the following [6], [3], [7]:

- Token Distribution: performing sub-words tokenization (e.g. via Bert or gpt2 Large Language Model) and checking for the presence of too many "unknown" tokens can provide insights into the quality of a text input. The token distribution could also help understand the gross value of the text quality.
- Unicode Standard Analysis: analyzing text data (at character level) according to the Unicode Standard can helpful understand text quality, especially in multilingual contexts. This analysis can be applied at three different Unicode levels: character, script, and block.
- Readability Scores: readability formulas such as Flesch-Kincaid, Gunning Fog Index, or Coleman-Liau Index can be used to quantify the readability of the text. These scores indicate how easy or difficult the text is to read and understand.
- Grammar and Spelling Checking: identify grammatical errors, misspellings, and typos in the text.
- Semantic Analysis: use techniques such as sentence embeddings to measure the semantic coherence of the text. This could also involve assessing whether the meaning of the text is coherent and relevant to the topic.

The following is a list of some of the most widely used steps in the data preparation phase for textual data [6]:

- Tokenization: it could be different from the sub-word token representation in the data-quality assessment; traditionally it is implemented through word separation activity.
- Lowercasing: lowercase the words.
- Removal of special characters: e.g. punctuation, extra spaces, tabs, and newlines (remove everything is a nonprinting character).
- Contraction expansions: replace abbreviations with their long form e.g., it's -> it is, won't-> will not.
- Stop-word removal: remove the common words that are often used in speech or text that may not be wanted to be included in the final analysis.
- Spell corrections: e.g. hapening-> happening
- Stemming: stemming is a heuristic process that chops off the ends of words to remove affixes (prefixes, suffixes, infixes) and reduce words to their stems. It applies rules (e.g. Porter algorithm) to remove common suffixes or prefixes, but they may not always produce a valid root word. The resulting stem may not be a real word. As an example, the word "running" stems to "run".
- Lemmatization: map words to their base or dictionary form, known as the lemma (root form). It typically involves dictionary lookup, morphological analysis, and knowledge graphs (e.g. WordNet), to properly reduce words to their base form. It results in valid words, which makes it preferable for tasks where word meaning and context are important. It requires more effort than stemming and is slower. As an example, the word "better" lemmatizes to "good".

Once the textual data is properly processed via such methods, it is common to utilize some of the following techniques for feature extraction and transformation into numerical form.

Classical text mining features

- Bag-of-Words model: it stands out as the most straightforward method for converting textual data into vectors. Under this approach, every document is depicted as an N-dimensional vector, with N representing

the entirety of words within the processed corpus. Within the vector, each element signifies the existence or frequency of a particular word.

- TF-IDF: One of the most common issues related to the BoW model is that some words overshadowthe rest of the words, due to high frequency, as it utilizes absolute frequencies to vectorize the documents. The Term Frequency-Inverse Document Frequency (TF-IDF) model mitigates this issue by scaling/normalizing the absolute frequencies by the frequencies of the words and their inverse document frequence.

At this point text data is represented by vectors of numbers that can be managed by a machine learning algorithm such standard numerical features.

Embedding features

Nowadays, it is a common practice to apply Deep Neural Network models as automated feature extraction methods for text and image data. Typical examples are the so-called embedding models (e.g. Transformers architectures for text and CNN family for images) that perform feature extraction directly on raw data (extract Embedding Features), without the need for classical transformations that require a huge amount of time, effort, and domain understanding to be performed accurately. However, this approach does not exclude some standard preliminary phases concerning, for example, syntactic corrections, removal of special characters, and text enrichment, which are always good practices to be carried out. Some examples are word embedding features (e.g. from Word2vec, GloVe model) and sentence embedding features (e.g. from Doc2Vec, Sent2Vec, Bert model).

3.2.1 The information extraction pipeline for textual data

An information extraction (IE) pipeline aims to extract (semi)-structured data from unstructured data like text. The following techniques can be applied in addition to the previous standard methods and can be considered a set of advanced steps through a standardization and harmonization goal. This activity can be formally placed into the Data Preparation phase of the CRISP-DM methodology.

There are several ways to implement this sort of data-transformation pipeline that mostly depends on the final use case.

It could be composed, for example, by the following steps:

- Text Cleaning: address everything that came up in the data quality assessment and possibly extend it through some use case-specific steps (e.g. remove URLs, remove HTML tags, remove digits).
- Coreference Resolution: the task of finding all expressions that refer to the same entity in a sentence.
- Name Entity Recognition (NER): is the task that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.
- Entity Standardization: the task of standardizing some specific kind of information like dates and times, geographic coordinates, email addresses, etc.
- Named Entity Linking (NEL): is the task of assigning a unique identity to entities (such as famous individuals, locations, or companies) mentioned in text. It is different from NER because here the words of interest (names of persons, locations and companies) are mapped from an input text to corresponding unique entities in a target knowledge base. It is common to use knowledge bases derived from Wikipedia (such as Wikidata or DBpedia). Entity linking techniques that map named entities to Wikipedia entities are also called wikification.
- Relationship extraction: the task of determining the relationship between entities (often it is achieved using a DNN Transformer Architecture like Bert). This step is suitable for the creation of a knowledge graph with nodes and links that describe the entities and relations in an advanced semantic representation of information.

An overview of the IE pipeline for textual data is shown in the following figure.

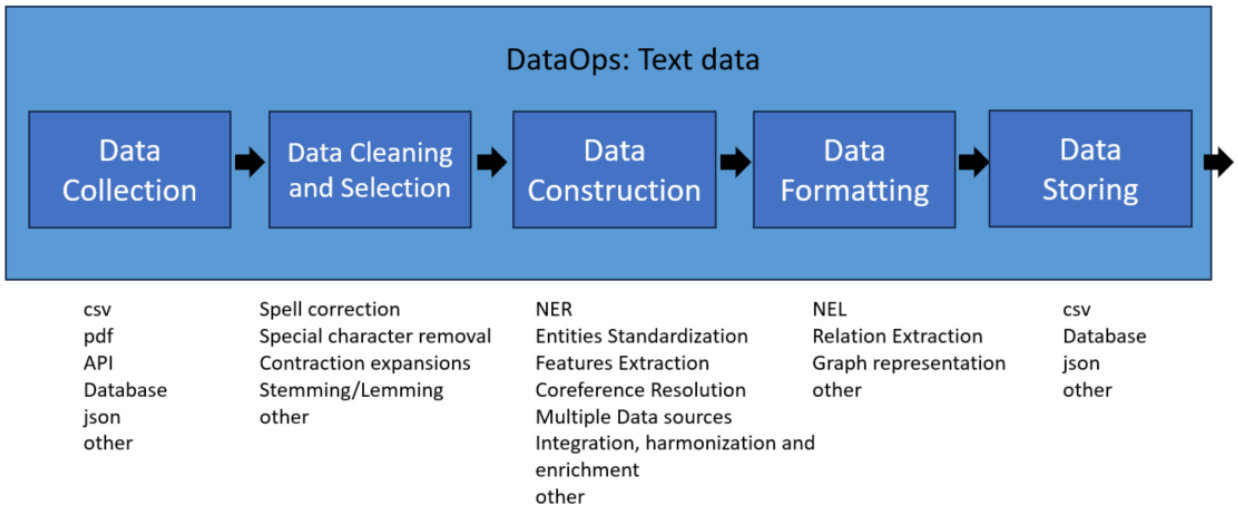

Figure 5. A possible DataOps pipeline for textual data

Those exposed are generic and standard process phases on textual data and must be adapted to the particular project at hand. In the CEDAR project, the definition and sequence of steps will be an iterative process which, adopting the principles of CRISP-DM, will try to improve over time.

3.3 Image Data

Images have an unstructured data format that requires specific controls and processing efforts to be considered in the DataOps pipeline.

In the following, there is a description of some common procedures to assess data quality and elaborate the images to produce more representative information for typical downstream tasks.

As with textual data, each project has its own peculiarities and it may be necessary to apply only some of these techniques, while the decision factors could vary from the initial data source type (from a camera, inside a pdf file, etc.) to specific project constraints (computational power, real-time requirement, etc.).

A list of techniques often used for the data-quality assessment in the image domain is the following [8]:

- Blobs detection: many images contain random pixels as noise (blobs); analyzing the distribution and characteristics (such as their size, shape, and intensity) of the detected blobs can provide insights into the level of noise present in the image.
- Centrality of the scene: the image may not be centered and may contain large black borders, and therefore it needs to be pre-processed. The analysis consists in the calculation of the bounding box of the largest contour and comparing its dimensions with the dimensions of the entire image.
- Color composition: analysis of the distribution of the RGB channels colors. Very skewed distributions can emerge from anomalous images. Overall, analyzing the RGB composition of an image provides valuable insights into various aspects of its quality, including color balance, saturation, contrast, and vibrancy.
- Pixel intensity distribution: analysis of the distribution of the pixel intensities often in a black-and-white representation. The shape of the pixel intensity histogram can provide valuable information about image quality. Like in the previous case, very skewed distributions are symptomatic of bad-quality images and histograms with spikes or gaps might suggest issues such as uneven illumination or sensor artifacts. In contrast, a well-distributed histogram with a peak in the middle may indicate good overall image quality.

- Dynamic Range: the dynamic range of an image, which is the range between the darkest and brightest pixel intensities, can also be inferred from the pixel intensity distribution. A wide dynamic range suggests good image quality, while a narrow range may indicate limited detail in either shadow or highlight areas.
- Noise: noise can affect the pixel intensity distribution, resulting in random fluctuations. By analyzing the distribution, one can identify the presence of noise, which is crucial for assessing image quality and determining appropriate noise reduction techniques.

Based on the outputs of the quality assessment the following steps could be implemented to clean and enhance image data:

- Blobs removal: remove the detected random small blobs on the image.
- Image cropping: remove the black background outside of the real image.
- Format standardization: add padding to preserve the aspect ratio and resize to target resolution are typical examples of standardization.
- Image Denoising: noise reduction is crucial for improving image quality. Techniques such as Gaussian blur, median filtering, or bilateral filtering can be applied to remove noise while preserving edges.
- Super-Resolution: super-resolution techniques enhance the resolution of an image, effectively increasing its quality. Methods include single-image super-resolution (e.g., using deep learning models) and multi-image super-resolution (e.g., using techniques like image fusion).
- Image Restoration: this involves recovering the original image from a degraded version. Deblurring techniques can be used to reduce blur caused by motion or defocus. Inpainting methods can fill in missing or damaged parts of an image.
- Edge Detection and Enhancement: identifying and enhancing edges in an image can improve its perceptual quality. Techniques like Canny edge detection or edge-preserving smoothing can be used.
- Image Dehazing: Removing haze or fog from images can significantly improve visibility and image quality. Dehazing algorithms estimate and remove the atmospheric veil from images using different models and priors.

As mentioned before in the context of textual data it is common to utilize some of the following techniques for feature extraction and transformation into numerical form.

Classical images features

This list can enumerate traditional image feature extractors like edge detector, color and texture descriptors, DoG Detector, SURF detector, SIFT Features, etc. [9]. A complete description of these techniques is out of the scope of this paragraph and probably some of them will be introduced and described in other Cedar deliverables.

Embedding features

Even in the context of images, it is common today to use Deep Neural Network models as automatic feature extractor methods. Some approaches use Convolutional Neural Networks (CNNs), autoencoders, ViT transformers, Generative Adversarial Networks (GANs), etc. [10].

This feature extraction step still benefits if carried out after the cleaning process described above. As usual, a good cleaning process of the input data is the basis of a good final result.

3.4 Tabular Data

With tabular data, it is probably easier to manage the preparation steps and of course the cleaning step, as it is a structured data type.

The following analysis can be applied also to time series data, which is a special case of tabular data with a timestamp column. Some additional advice and restrictions should be taken into consideration due to the natural causality of time series data.

Generally, dirty tabular data is usually presented in two forms: missing data or anomaly/outlier data.

There is also the case of duplicated data. This case is simple to manage, and the advice is to always remove duplicates because they do not give any contribution to the machine learning algorithm and there is the risk that they fall in both the training and the validation (test) set, biasing the real model's performance evaluation.

The ways of handling missing values and anomalous data are quite different:

- The instances containing missing data can be ignored, filled in manually with a constant, or filled in by using estimations over the data.
- For anomalies, some statistical and descriptive techniques can be used to identify outliers and filters can be applied to handle them.

Handle Missing Data

There are two most used ways of dealing with missing data [6]:

- *Dropping observations*: this is the simplest approach and consists in removing entire columns (features) or rows (entries) from the dataset, if they contain empty fields. Dropping missing values is sub-optimal because dropping observations causes drops of information. In real world applications, it is very frequent the need to make predictions on new data even if some of the features are missing. Thus, the drop implementation in the training phase can lead to issues in the inference phase, making predictions impossible. This situation should be avoided as much as possible. Additionally, the fact that the value is missing might be informative of something harmful in the actual data pipeline.

- Imputing the missing values:
  Missing categorical data: the best way to handle missing data for categorical features is to simply label them as 'Missing', essentially adding a new class for the feature that will be used by the algorithm accordingly.

  Missing numeric data:
  - filling with a default special value (it could depend on the specific domain application).
  - impute missing values via the feature means, medians, mode, etc. (univariate intra-features imputation). An example is the sklearn library's SimpleInputer class [11]. For time series data, it is better considering some points close to the missing one and not the entire dataset for that feature.
  - allowing an algorithm to estimate the missing values using the entire set of available features (multivariate inter-features imputation). Examples are the IterativeImputer and KNNImputer classes in the sklearn library [11]. A general approach is to build a regressor or a classifier (with the no missing value feature set) for computing the value to impute. For time series data the advice described for intra-features is still valid.

There is no rule or best practice for handle missing data, and the choice of the appropriate missing data imputation method depends on score results, use case judgement, and domain knowledge.

Handle Outlier

An outlier can be defined as: "A data object that deviates significantly from the normal objects as if it were generated by a different mechanism".

Outliers are interesting because they violate the mechanism that generates the normal data, so their management has to be taken into account. As usual, for some scenarios, it can be useful to drop the column (row) where anomalies are detected. In others, anomalies can be replaced with imputed values. Finally, in certain scenarios, it is crucial to leave them as they are.

As already seen for missing data, outliers can be managed in a univariate approach (e.g. looking at the position of the point with respect to the interquartile range IQR, Grubb's test) or in a multivariate approach considering all the features in the dataset.

Outlier detection (and the related novelty detection task) is a wide topic that needs a dedicated chapter to be properly covered. Here we showed only some common techniques, while the details can be found in the reference notes.

- Statistical methods (also known as model-based methods): assume that the normal data follows some statistical model (a stochastic model e.g. a gaussian distribution) and data points not following the model are outliers.
- Proximity-Based Methods: an object is an outlier if the nearest neighbours of the object are far away, i.e., the proximity of the object significantly deviates from the proximity of most of the other objects in the data set.
- Reconstruction error approach: using dimensionality reduction techniques (like PCA and autoencoder) is possible to "compress and expand" the feature set keeping the reconstruction error under a certain threshold for usual instances while the same error will be higher for uncommon instances that can be categorized as anomalies.

Feature engineering

Feature engineering refers to the task of transforming raw data into features that better represent the underlying problem to the predictive models, thereby improving their performance. It involves creating new features from existing data or modifying existing features to enhance the predictive power of machine learning algorithms.

Examples of feature engineering include:

- *One-Hot Encoding:* converting categorical (pure nominal) variables into binary vectors, where each category becomes a separate feature with a value of 1 or 0. It is also known as the dummy-variables approach.
- *Feature Interactions:* creating new features by combining existing features. For example, multiplying two numerical features together or combining categorical features to capture combined effects.
- *Polynomial Features*: generating higher-order polynomial features to capture nonlinear relationships between variables.
- *Binning or Discretization*: grouping continuous features into bins or discrete intervals (lower range of possibilities), which can help capture nonlinear relationships and reduce the impact of outliers.
- *Time Features:* extracting information from timestamps such as day of the week, month, hour, or season, and generating a column for each of those new features. This technique, analogous to the one-hot encoding approach, helps in the time series forecasting task revealing temporal patterns in the data. Under the same domain, other techniques are *the cyclical encoding with sine/cosine transformation* and the use of *radial basis functions* [12].

Data transformation

Data transformation includes activities to address distribution issues like strong asymmetry and presence of peaks. This task aims to reduce these issues by defining transformations that preserve the relevant information, eliminate at least one problem on the initial dataset, and finally make the dataset more useful.

The goal is twofold:

Main goals:

- Stabilize the variances
- Normalize the distributions
- Make linear relationships among variables

Secondary goals:

- Simplify the elaboration of data containing problematic features
- Represent the data in a scale considered more suitable (all characteristics represented in the same scale range).

The main reason behind this effort is that many statistical methods require linear correlations, normal distributions, and the absence of outliers, to perform at their top level of performance.

Many data mining algorithms can automatically treat non-linearity and non-normality, but usually they work better if such problems are treated beforehand. A typical example is the Neural Network approach where it is quite mandatory to normalize the features at the very early stage of the process.

In general, if the algorithm embraces a gradient-descent approach, it works much better (will be more robust, numerically stable, and converge faster) if the data is centered and has a smaller range.

There are many ways for scaling the features. The most common "normalization" schemes are:

- Min-Max scaling: it squashes the features into a [0, 1] range.
- Z-Score standardization: after standardizing a feature, it will have the properties of a standard normal distribution, that is, unit variance and zero mean. Notice: this does not transform a feature that does not follow a normal distribution into one that does.

Other less frequent transformations include [6], [3]:

- Normalization by decimal scaling
- Exponential transformation
- Logarithmic transformation

Dimensionality Reduction

Some purposes of dimensionality reduction techniques are the following [3]:

- Avoid curse of dimensionality (the effect when dimensionality increases, data becomes increasingly sparse in the space that it occupies, causing the definitions of density and distance between points become less meaningful)
- Reduce amount of time and memory required by data mining algorithms (computational efficiency)
- Allow data to be more easily visualized and interpreted
- May help to eliminate irrelevant features or reduce noise (easier data collection and management)

It is possible to derive two branches: the feature selection branch, where the methods aim to select a sub-set from the initial set of features without altering them; and the features extraction one, where the original features are transformed in something new.

Feature extraction: this set of methods derives a new set of features from the original ones. The transformation is irreversible.

- Principal Component Analysis (PCA): a linear not supervised technique with the goal of finding a projection that captures the largest amount of variation in the data (it finds the eigenvectors of the covariance matrix).
- Singular Value Decomposition (SVD): a linear supervised technique that tries to maximize the patterns discrimination.
- t-distributed Stochastic Neighbor Embedding (t-SNE): is a non-linear and unsupervised technique useful for representation purposes.
- Uniform Manifold Approximation and Projection for Dimension Reduction (U-map): is similar to t-SNE with some additional benefits.
- Embeddings: a compact representation of information extracted through a Neural Network. This form of extraction is typical for images, texts, and graphs, but could also be used with tabular data and timeseries data.

Feature selection: this set of methods tries to minimize the set of features to use in the problem, eliminating irrelevant and redundant features. A performance indicator must be taken into consideration to judge the quality of the resulting selection.

- Brute-force approach: tries all possible feature subsets as input to the data mining algorithm and selects the combination that gives the best performance. This exhaustive way is expensive and impractical with a large number of features.
- Embedded Methods: feature selection occurs naturally as part of the data mining algorithm. An example is the Random Forest feature importance value.
- Filter Methods: features are selected before the data mining algorithm is run. Some examples are the variance threshold, Information gain, correlation with target, and pairwise correlation methods [13].
- Wrapper Methods: they use the data mining algorithm as a black box to find the best subset of attributes. Common approaches include Recursive Feature Elimination (RFE), Sequential Feature Selection (SFS), and Permutation importance methods [13].

3.5 DataOps for AI-oriented Knowledge Graph

Based on the descriptions of the data sources listed in the deliverable D2.1 and the need to adopt the concept of semantic data models in compliance with the IDSA standards, the following figure shows the entire pipeline designed for the CEDAR project.

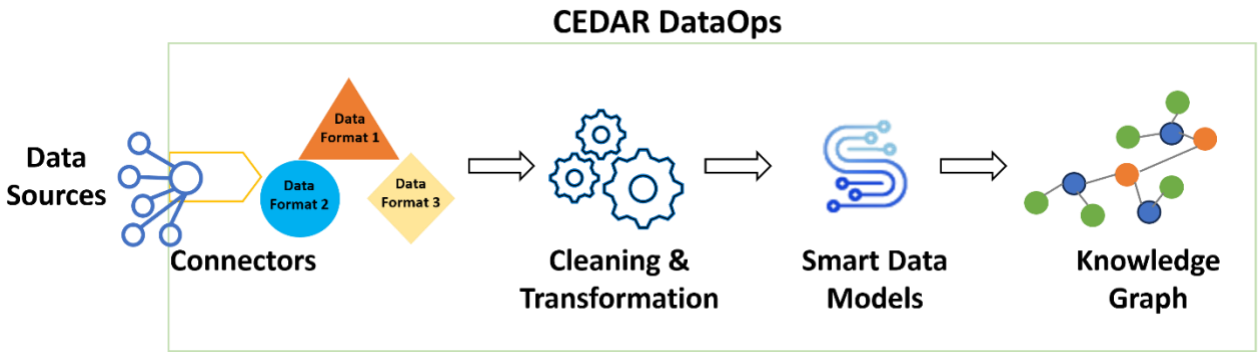

Figure 6. CEDAR DataOps pipeline

The various data sources will be ingested with dedicated technologies and protocols (restAPI, file copy, etc.), then the most suitable cleaning and transformation processes will be carried out until the generation of semantic data structures, the harmonized representation based on smart data models. This representation will be finally transformed into a graph semantic and stored into an appropriate system such as a graph DB.

Once the graph is stored in a graph DB, any other application will be able to extract information. For example, it will be possible to apply a machine learning algorithms or a graph neural network (GNN) to the graph. The graph representation can help identify patterns, relationships, and insights that may not be evident in the original dataset format.

3.5.1 Smart Data Models

The Smart Data Models (SDM) [14] is a collaborative initiative aimed at promoting the adoption of a reference architecture and standardized data models to support a digital market of interoperable and replicable smart solutions.

Lead by The FIWARE Foundation, IUDX, TM Forum, OASC, and others, the initiative aims to support a digital market of interoperable and replicable smart solutions across various sectors, beginning with smart cities.

The widespread adoption of de facto standard information models is crucial for establishing a global digital single market of interoperable and replicable smart solutions across various domains, such as smart cities, smart agrifood, smart utilities, and smart industry. These models are essential for creating the common technical foundation necessary for standard-based open innovation and procurement.

“Data Models play a crucial role because they define the harmonized representation formats and semantics that will be used by applications both to consume and to publish data” [14].

The CEDAR project, with its foundations in the data-space solution, is aligned with this need for data standardization and harmonization in order to facilitate and streamline data exchange between the various actors.

Several potential CEDAR’s relevant SDM are already defined:

- Organization
- Person
- Public Accountability
- Data Quality
- Social Media
- Sustainable Development Goals

However, the absence of a formally defined SDM does not preclude the application of the same principle on data semantics deemed more appropriate during the project; indeed, it will be possible to define similar structures for custom data specific to the CEDAR project.

A common format to represent smart data models is the JSON linked data (LD) format. JSON, as defined in RFC7159, is a straightforward language for representing objects on the Web and Linked Data describes content across various documents or websites. JSON-LD aims to provide a simple way to publish Linked Data in JSON format and to add semantics to existing JSON. It is also valuable for building interoperable web services and for storing Linked Data in JSON-based document storage systems (e.g. NoSQL DB and Data Lake).

Upon this concept is the "Next Generation Service Interfaces" NGSI-LD format that represents the standardized ETSI (European Telecommunications Standardization Institute) version. In a nutshell NGSI-LD uses JSON-LD as its base format to leverage its powerful linked data capabilities, while adding domain-specific extensions and features to

support the needs of data harmonization. This combination ensures that data can be semantically enriched, interoperable, and effectively managed in complex systems.

A Smart Data Model can be defined through a NGSI-LD format [14] in both keys-values and normalized format.

The following figure shows an example of NGSI-LD for Organization entities.


```
{
    "id": "urn:ngsi-ld:Organization:34f91f29-aadd-45f7-ab9e-4fca2baffdd7",
    "type": "Organization",
    "dateCreated": {
        "type": "Property",
        "value": "2022-06-21T08:24:35.905712+02:00"
    },
    "dateModified": {
        "type": "Property",
        "value": "2022-06-22T09:24:35.905712+02:00"
    },
    "name": {
        "type": "Property",
        "value": "Example Organization"
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "type": "Point",
            "coordinates": [
                49.4,
                8.68
            ]
        }
    },
    "address": {
        "type": "Property",
        "value": {
            "addressLocality": "Heidelberg",
            "postalCode": "69115",
            "streetAddress": "Example-Street 42"
        }
    },
    "areaServed": {
        "type": "Property",
        "value": "Stadt Heidelberg"
    },
    "url": {
        "type": "Property",
        "value": "https://www.example-organization-homepage.com"
    },
    "legalName": {
        "type": "Property",
        "value": "Beispielname GmbH"
    },
    "taxID": {
        "type": "Property",
        "value": "12345678900"
    },
    "@context": [
        "https://smart-data-models.github.io/DataModel.Organization/context.jsonld",
        "https://raw.githubusercontent.com/smart-data-models/dataModel.Organization/master/context.jsonld"
    ]
}
```


Figure 7. NGSI-LD (Normalized) Smart Data Model for Organization

In the JSON shown, the first line is a unique identifier for the organization, using a URN (Uniform Resource Name) that follows the NGSI-LD specification; the @context key defines an array that provides URLs to JSON-LD context definitions that help interpret the data model. These context URLs define the semantics of the terms used in this JSON object, ensuring interoperability by providing machine-readable definitions of each attribute.

The NGSI-LD format can represent graph networks by representing data as entities with unique identifiers. These entities can have properties and relationships to other entities, forming a graph of interconnected data points.

It the following figure is shown a graph structure with the following entities and relations:

- Device: A temperature sensor located at a specific location and measuring temperature.
- Location: A specific address where the device is located.
- Measurement: The value and unit of the temperature measurement.

```
"id": "urn:ngsi-ld:Device:1",
"type": "Device",
"name": {
  "type": "Property",
  "value": "Temperature Sensor"
},
"locatedAt": {
  "type": "Relationship",
  "object": "urn:ngsi-ld:Location:1"
},
"measures": {
  "type": "Relationship",
  "object": "urn:ngsi-ld:Measurement:1"
}
,

"id": "urn:ngsi-ld:Location:1",
"type": "Location",
"address": {
  "type": "Property",
  "value": "123 Main St, Springfield"
}
,

"id": "urn:ngsi-ld:Measurement:1",
"type": "Measurement",
"value": {
  "type": "Property",
  "value": 22.5
},
"unit": {
  "type": "Property",
  "value": "Celsius"
}
```


Figure 8. NGSI-LD (simplified) representation of a graph structure with multiple types of entities and relationships

3.5.2 Knowledge Graph Generation

The first step to generate a knowledge graph (KG) is to define the scope and purpose of the knowledge graph itself. This starts by identifying the type of information to include (nodes) and the specific relationships and connections that will be represented in the graph (edges) where nodes and edges could have their specific features and properties.

For some problems, building the graph to represent the overall semantics could become a complex operation and it is better to follow heuristics to rationalize the various steps.

A common approach is starting from a Minimum Viable Graph (MVG) and repeat these steps:

- Extract: identify and load relevant data source
- Enhance: empower the data (e.g. extract embedding and use them as a new node attribute)
- Expand: link related information to extend the context

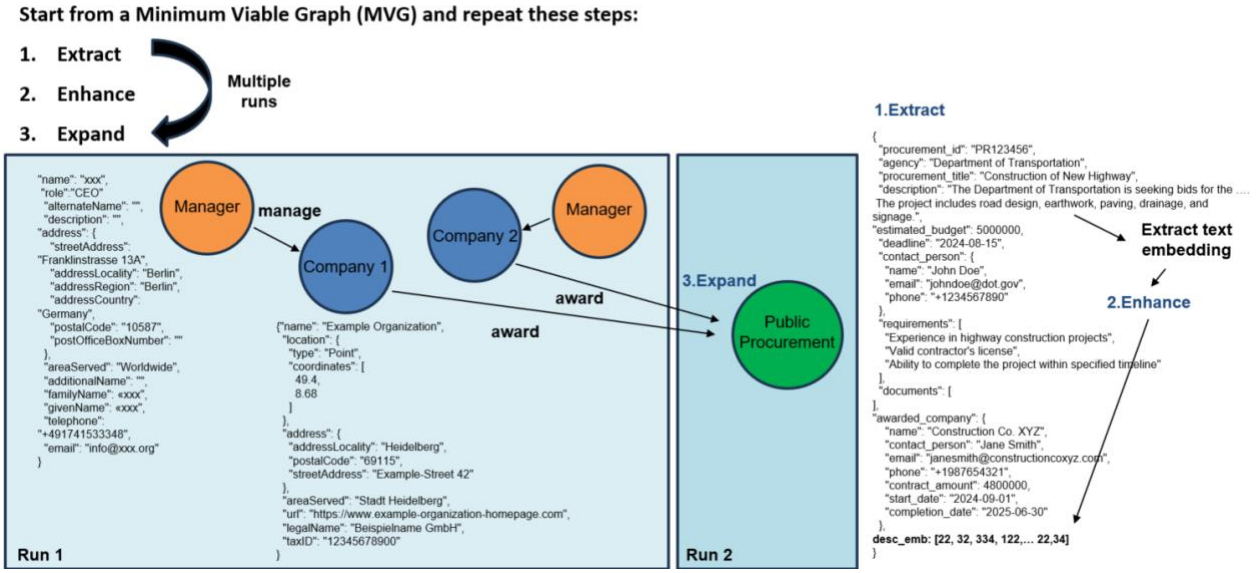

Figure 9. Knowledge Graph generation

As an example, at run 1 the extraction phase can be associated with the reading of documents that describe isolated entities; the subsequent enhance phase could be the association of the smart data model and the expansion phase could be the link between the different entities by means of attribute matching. At run 2 the textual contents of the individual attributes could be transformed into the corresponding embedding by applying a deep neural network enhancing the feature representation or perhaps a NER model could be applied to extract new entities and therefore be able to expand the graph topological structure by generating new nodes and edges.

A cycle methodology like this can help both in achieving the desired final graph but also in keeping track of the evolution of the graph in subsequent cycles, avoiding aiming for the final graph in a single step.

Nowadays among the tools that can assist in these Extract, Enhance and Expand phases is the use of Large Language Models. An exhaustive analysis of these methodologies is outside the scope of this document, please refer to the paper in the references for details [15].

3.6 DataOps Technologies and Tools

As discussed in the previous paragraph, a DataOps pipeline typically involves tasks such as data extraction, transformation, loading, orchestration, monitoring, and management. There are several tools that offer a wide range of features for building robust DataOps pipelines, and the choice depends on specific requirements, preferences, and constraints of the project.

Data processing architectures can be broken into two main categories [16]:

- Batch: a job begins either upon request or at a scheduled time, fetches and processes data, and writes the results to the target storage upon completion. Batch jobs typically require more time to process.
- Stream: continuous processing of incoming requests or data chunks, with results written in real time to target storage or a message queue.

Batch processing is generally more efficient for handling large quantities of data where processing time is less of a concern. However, interactive and stream data processing provide faster responses with shorter delays. Additionally, constructing data stream processing pipelines is typically more complex than creating batch jobs.

A common application of batch processing is in ETL tasks. ETL, which stands for Extract, Transform, Load, involves extracting data from various sources, transforming it, and loading it into a target database, data warehouse, or data lake. ETL is essential for data integration, enabling organizations to extract, clean, and transform data from multiple sources into a single, centralized repository.

Some frameworks can support both processing methods while others are optimized to work in only one modality.

In the following will be described some of the most used solutions in the context of DataOps and potentially suitable for the CEDAR project which, due to its complexity, requires a well-engineered dedicated framework able to:

- Define complex data-pipelines
- Monitor the pipelines
- Manually trigger or schedule the pipelines
- Smart resource management
- Full control from external tools (e.g. via REST API)

3.6.1 Apache Airflow

Airflow is an open-source platform to programmatically author, schedule, and monitor workflows. It is widely used for orchestrating complex data pipelines. Airflow allows you to define workflows as Directed Acyclic Graphs (DAGs) of tasks where each task represents an individual processing step. The tasks can be written in Python and executed in diverse environments, such as local setups, Kubernetes clusters, or cloud platforms.

Airflow provides a web UI interface (fig. 10) from which it is possible to monitor each workflow, start it manually, and set the scheduling policy.

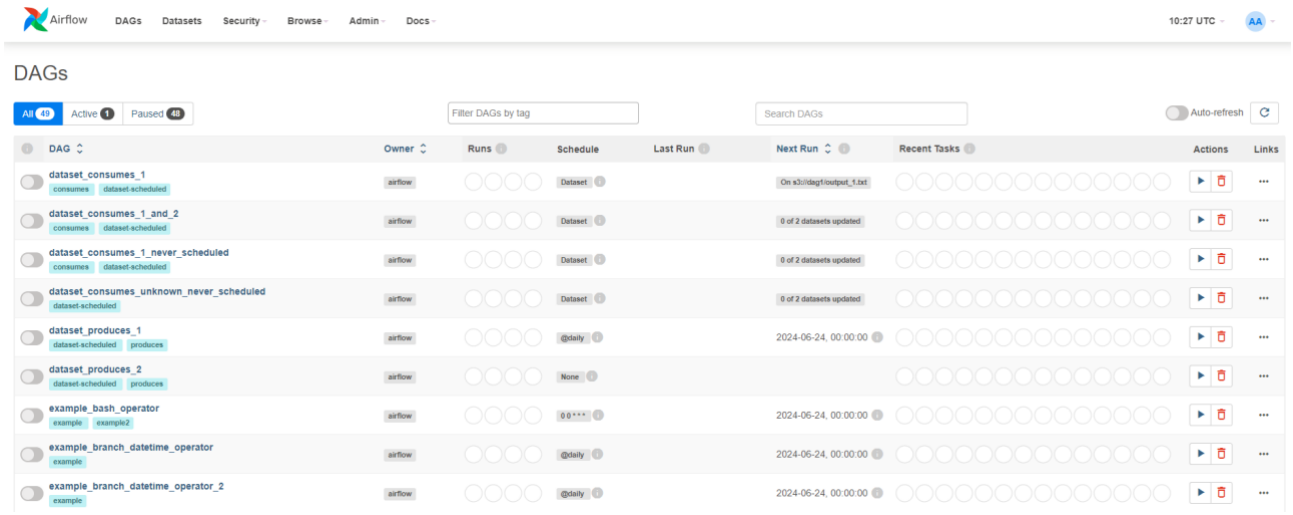

Figure 10. Airflow DAGs overview

The calendar view (fig.11) allows a quick overview of the history of past processes carried out and their outcome, as well as viewing future scheduled executions.

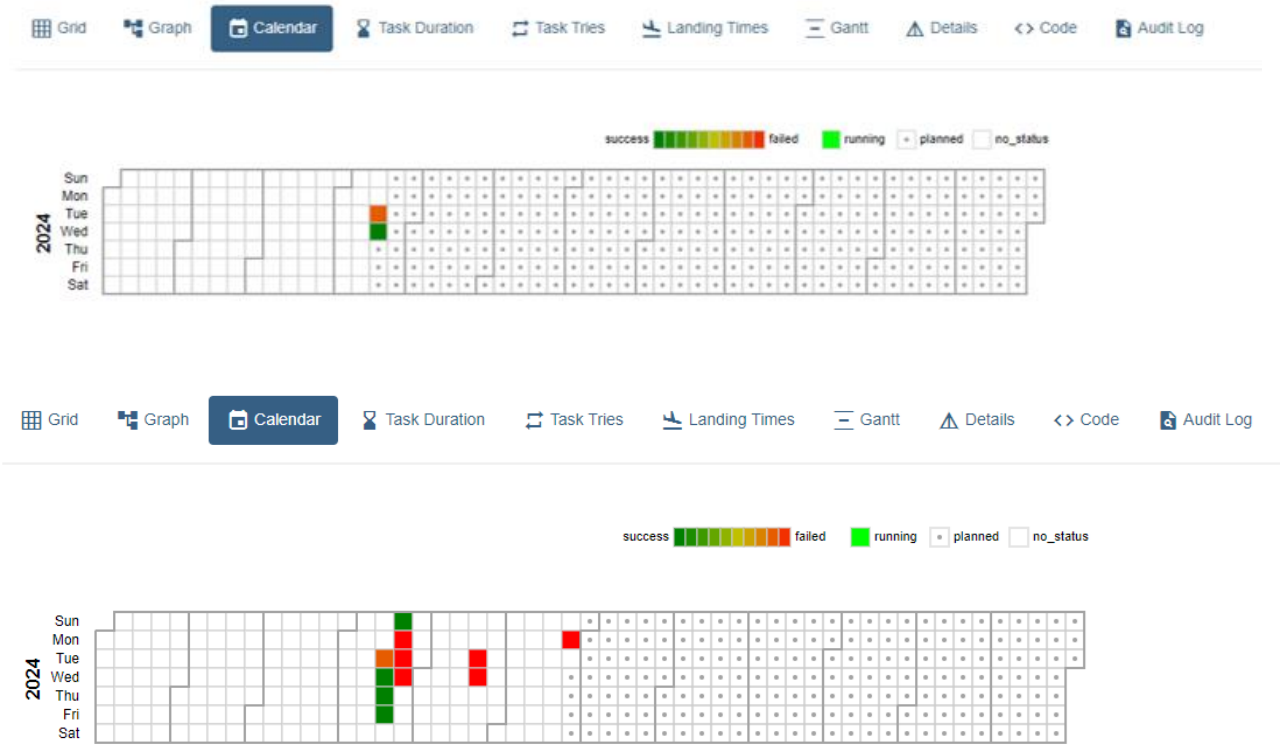

Figure 11. Airflow calendar view

Within a single DAG it is also possible to monitor in real-time the status of each single task (green: success, light green: running, red: a problem happened).

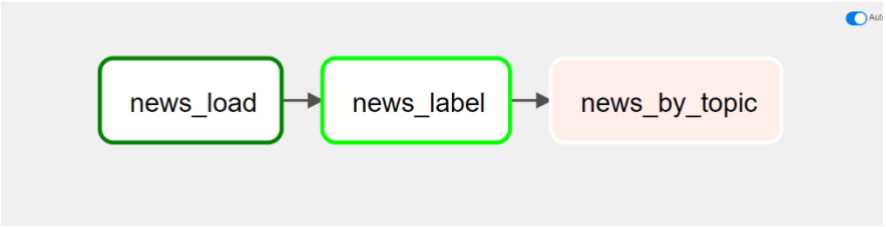

Figure 12. An example DAG with three tasks

The tasks within a DAG can be defined at the Python code level or it is possible to invoke external services (for example services that interact via REST API) that carry out the operation or even associate the execution of an autonomous pod [66] managed via parameters to a command line interface (CLI).

The coordination of the computational load of the DAGs is handled through the instantiation of dedicated pods that optimize the strategy of machine resources. At the setting level, it is possible to define in detail all the load management policies such as the number of pods associated with the DAG, the resources for each activity, etc.

Another good feature of Apache Airflow is the full control through a REST API interface. Using the API specifications, it is possible to monitor DAGs status, trigger a DAG, list events log, etc. This enables the framework to be integrated and managed from other tools while ensuring all standard functionality.

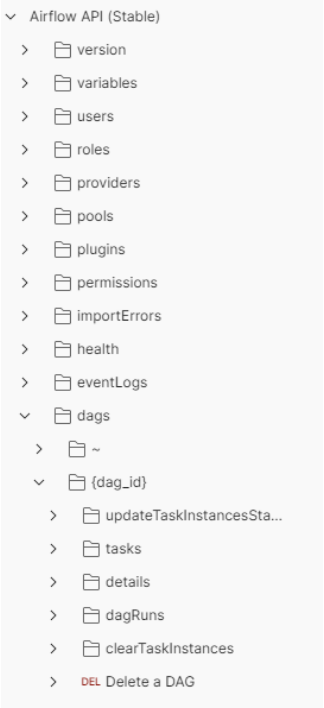

Figure 13. Airflow REST API collection example

The only limitation of using this technology is that it is not suitable for the stream processing mode. To cite verbatim what is specified in the official documentation "While the CLI and REST API do allow triggering workflows, Airflow was not built for infinitely running event-based workflows. Airflow is not a streaming solution." [17]

From discussions with the CEDAR pilot's representatives, it seems that the processing stream mode is neither required nor scheduled, and user-triggered modes may be sufficient.

Considering all these aspects, it can be stated that this technology fully respects the requirements and needs of the CEDAR project.

3.6.2 Apache Beam

Apache Beam is an open-source, unified model for defining both batch and streaming data processing pipelines [18], no matter if on-prem or in the cloud.

A good feature is that it is possible to write the pipeline in several programming languages like Python, Java, GO, Scala, etc. widening the possibilities for development by programmers of different backgrounds.

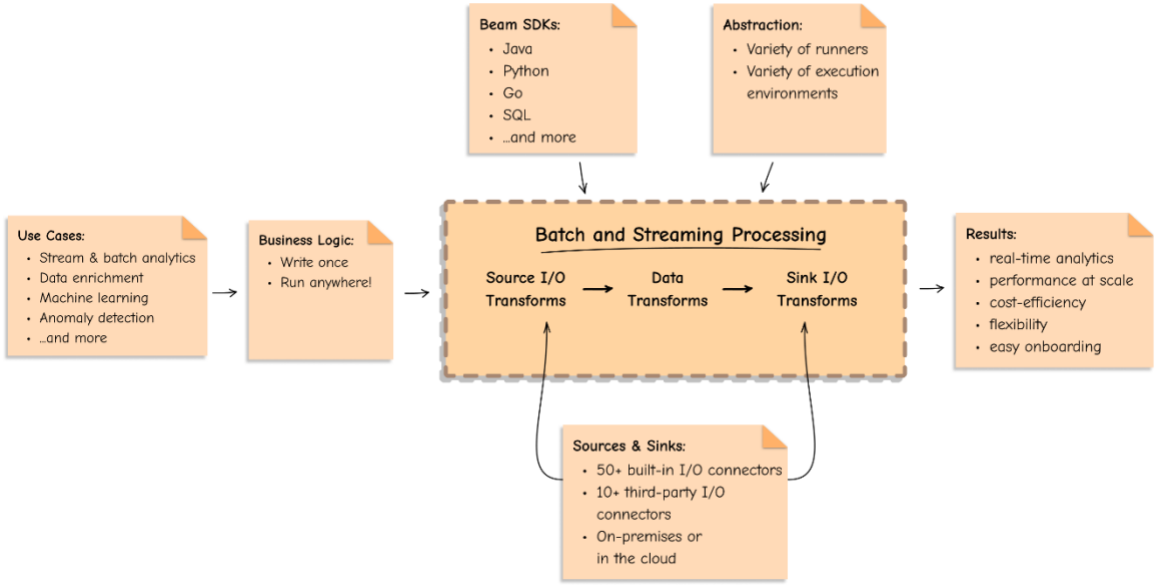

Figure 14. Apache Beam main features and capabilities [18]

The current release 2.56.0 has two major drawbacks:

- It does not provide a User Interface by default and everything (logs, status, etc.) has to be managed by code.
- It does not provide a REST API or other kinds of interfaces and everything remains in the execution flow.

These two limitations could have a large limiting impact within the project and could weigh much more than the advantages that the framework guarantees.

3.6.3 Apache Flink

Apache Flink is an open-source framework that can be used in tasks that require batch mode but also stream mode [19]. It is designed specifically for low-latency event processing, making it suitable for real-time analytics and event-driven applications but it can also handle finite datasets, allowing batch processing as well, making the implementation of Extract, Transform, and Load (ETL) processes possible. Another key feature is the high throughput; indeed, the engine is optimized to process large volumes of data with high-speed demand.

Several programming languages (Python, Java and Scala) can be used to define the execution steps, allowing its adoption by a wide audience of software developers.

Like Airflow Flink, it provides a REST API specification that enables full control from external services. It provides also a web UI, even though, at the current release (1.19.0), it is not particularly advanced as the one provided by Airflow.

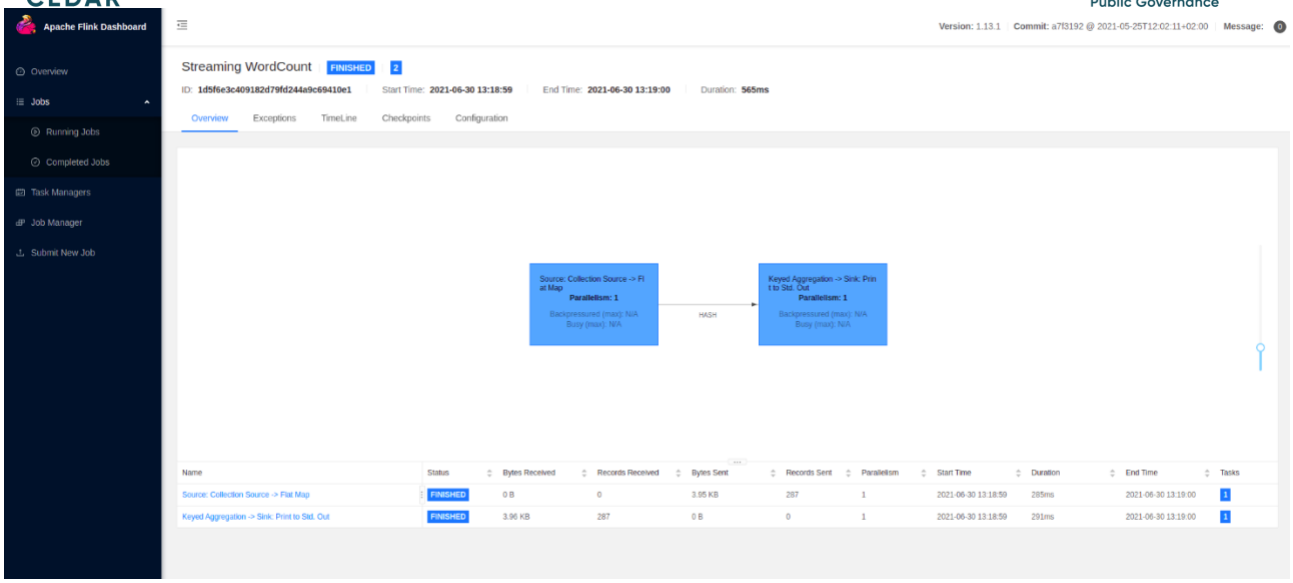

Figure 15. Apache Flink web UI [19]

All the features described make the framework very powerful and, at the same time, extremely flexible. The final choice of adoption in the CEDAR project depends not only on purely technical factors but also on other factors such as the availability of updated documentation and the presence of an active community.

3.6.4 Apache NiFi

Apache NiFi is a data processing and integration open-source platform that provides a visual interface for designing, controlling, and monitoring data flows.

The default programming language is Java, but it is possible to develop logical blocks in other programming languages like Python.

Its main pupose is stated directly on the official web site: “Put simply, NiFi was built to automate the flow of data between systems. While the term 'dataflow' is used in a variety of contexts, we use it here to mean the automated and managed flow of information between systems” [20].

It offers a lot of important features in the context of data transfer tasks like, for example, Data Flow Management, Data Provenance tracking, data source ingestion and Data Transformation and Enrichment blocks. Security is also a relevant topic. Indeed, Apache NiFi provides robust security features to ensure the confidentiality, integrity, and availability of data. It supports authentication, authorization, encryption, SSL/TLS, data masking, and other security mechanisms to protect sensitive information.

It allows batch and stream data processing: the flows can be scheduled or triggered through a message from a stream (e.g. a Kafka topic).

NiFi offers comprehensive monitoring and management capabilities to track the performance, health, and status of data flows in real-time, everything through the native web interface or the dedicated REST APIs.

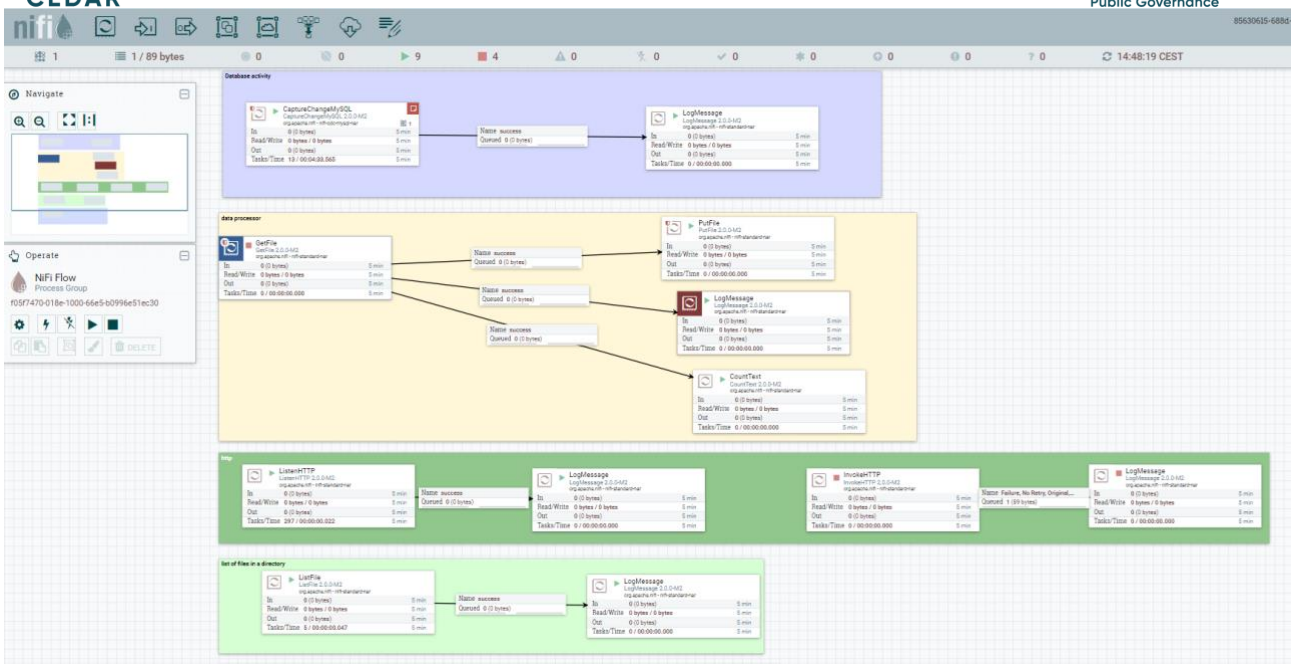

Figure 16. Apache NiFI flows example

One of the disadvantages of the technology compared to Airflow is that the UI interface does not describe in detail all the information on the status of the individual blocks that make up the flow nor is there a calendar view of the scheduling of the pipelines with the respective outcomes.

Another drawback is the fact that it does not support granular management of individual operational blocks, in the sense that it is not possible to use a dedicated pod for a specific task. This also impacts the management of the computation load which can only be scaled horizontally.

3.6.5 Final considerations

Given the specific characteristics of the CEDAR project where stream processing is not required and given the importance of features like user control and flexibility of the solution, the direction for the DataOps backbone technology would fall on the selection of Apache Airflow.

This choice does not preclude the use of other technologies in particular sub-activities or highly specific tasks, if vertical needs emerge during the project.

Starting from this concept, the next paragraph will introduce an Engineering proprietary tool, the Data Mashup Editor (DME), as a powerful and flexible solution that could be coupled with the core DataOps framework for speeding-up and streamlining several use case scenarios.

3.6.6 The ENG Data Mashup Editor

The ENG Data Mashup is an intuitive graphical no-code tool that simplifies the process of collecting and processing data from different sources, leveraging cutting-edge technologies and intelligent data integration techniques.

By connecting atomic modules via a web UI interface, it is possible to create complex data flows in a simple and intuitive way, guaranteeing visibility on the intermediate processing steps.

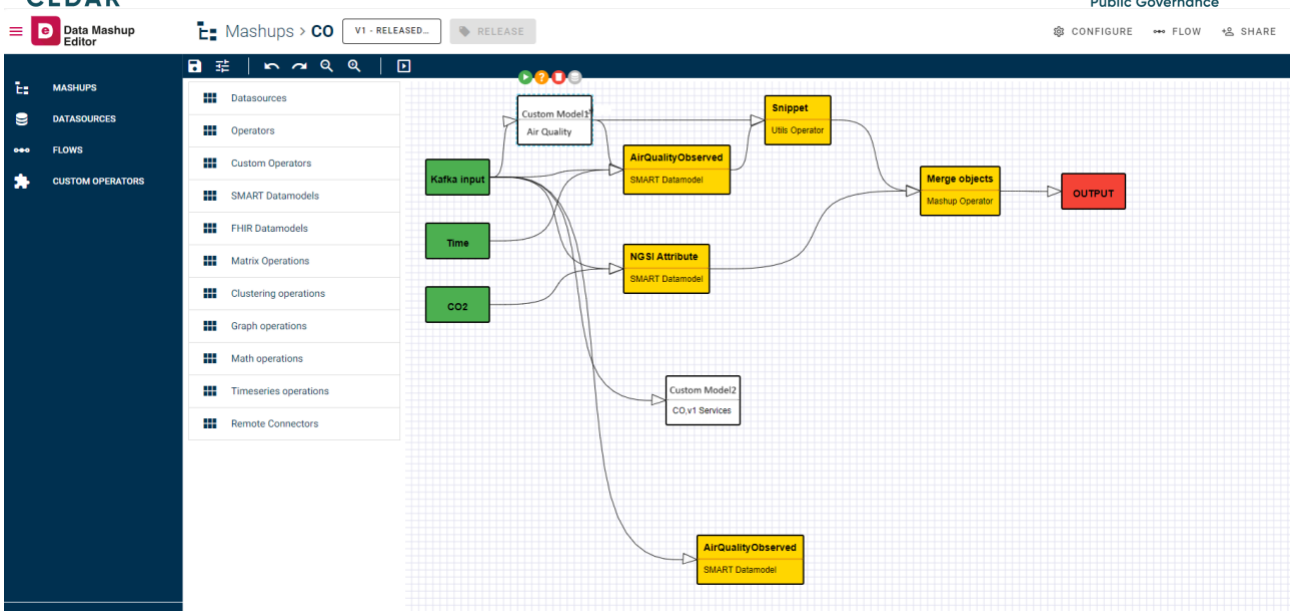

Figure 17. the Data Mashup Editor

Starting from data sources of different nature (e.g. Rest endpoints, Kafka messages, MQTT messages, etc.) the tool allows to transform, harmonize and extend the input data and return the elaborate output on various channels (HTTP, Kafka, MQTT).

The tool can support the CEDAR project in multiple ways. For example, since it does not require programming knowledge, it allows configurators of the pilot projects to independently manage some aspects related to data ingestion, guaranteeing the DataOps framework a coherent data format aligned with the predefined needs of the pipeline. Another advantage comes from the fact that some of the processing defined in the DataOps pipeline could be delegated to specific DME flows for efficiency needs, concretely lightening the burden on the execution of the main task.

In a nutshell, the DME can represent an element of additional flexibility to support the entire project needs.

4 MLOps Methodology

With the increasing integration of machine learning and artificial intelligence into various aspects of daily life and business operations, the efficient management and deployment of ML models have become essential. MLOps (Machine Learning Operations) addresses this need. This chapter provides an overview of MLOps, beginning with an "Introduction to MLOps". Next, the "Core Principles of MLOps" are examined to highlight the practices that ensure effective implementation. The section "MLOps Key Components" breaks down the essential elements required for a robust MLOps framework. Finally, "MLOps Technologies and Tools" explores the various technologies and tools that facilitate reliable and scalable MLOps practices, ensuring smooth transitions from development to production.

4.1 Introduction to MLOps

MLOps [21], a combination of "Machine Learning" and "Operations," involves a set of practices aimed at making the lifecycle of machine learning models more efficient and streamlined. The main goal of MLOps is to bridge the gap between data science and operational teams, enabling better collaboration and faster deployment of ML models. Traditionally, data scientists focus on creating and training models, while operational teams manage deployment and monitoring. MLOps brings these roles together, encouraging continuous integration and continuous deployment (CI/CD) specifically for ML. This approach addresses challenges unique to machine learning, such as handling large datasets, ensuring model reproducibility, and maintaining models in production environments. By implementing MLOps, organizations can become more agile, reduce time-to-market, and ensure their ML models perform consistently and reliably. As AI-driven insights become increasingly critical for maintaining a competitive edge, MLOps provides a framework for scaling and sustaining machine learning projects effectively.

Introducing MLOps into an organization can significantly improve how ML projects are managed. MLOps fosters a DevOps-like culture where data scientists, engineers, and operations teams work closely together, ensuring that models are both technically robust and practically viable in production environments. This collaboration reduces the friction often seen between development and operations, leading to smoother transitions from model training to deployment. Additionally, MLOps encourages the use of standardized processes and tools, which can reduce duplication of effort and improve the efficiency of the ML lifecycle. By focusing on automation, MLOps minimizes manual intervention, which not only speeds up the deployment process but also reduces the risk of human error, ensuring models are deployed consistently and reliably. Overall, MLOps is a transformative approach that enhances the robustness and scalability of ML projects, ensuring they deliver real value to businesses.

4.2 Core Principles of MLOps

The core principles of MLOps emphasize automation, collaboration, continuous improvement, and governance. Automation is central to MLOps, reducing manual tasks and minimizing the risk of human error through automated model training, validation, and deployment pipelines. Collaboration between data scientists and operations teams ensures models are technically robust and aligned with business goals and operational needs. CI/CD practices are fundamental, enabling rapid iterations and updates of ML models in response to new data and evolving requirements. These practices ensure models are frequently and reliably updated, decreasing the time from development to production. Additionally, principles of reproducibility and traceability are critical, allowing for consistent model replication and easy tracking of changes over time. Governance, encompassing security, compliance, and ethical considerations, ensures ML models adhere to regulatory requirements and organizational policies. Collectively, these principles form a robust framework supporting the sustainable and scalable deployment of machine learning models. These MLOps principles are listed in the Figure 18 along with the typical technical components found in a standard MLOps architecture.

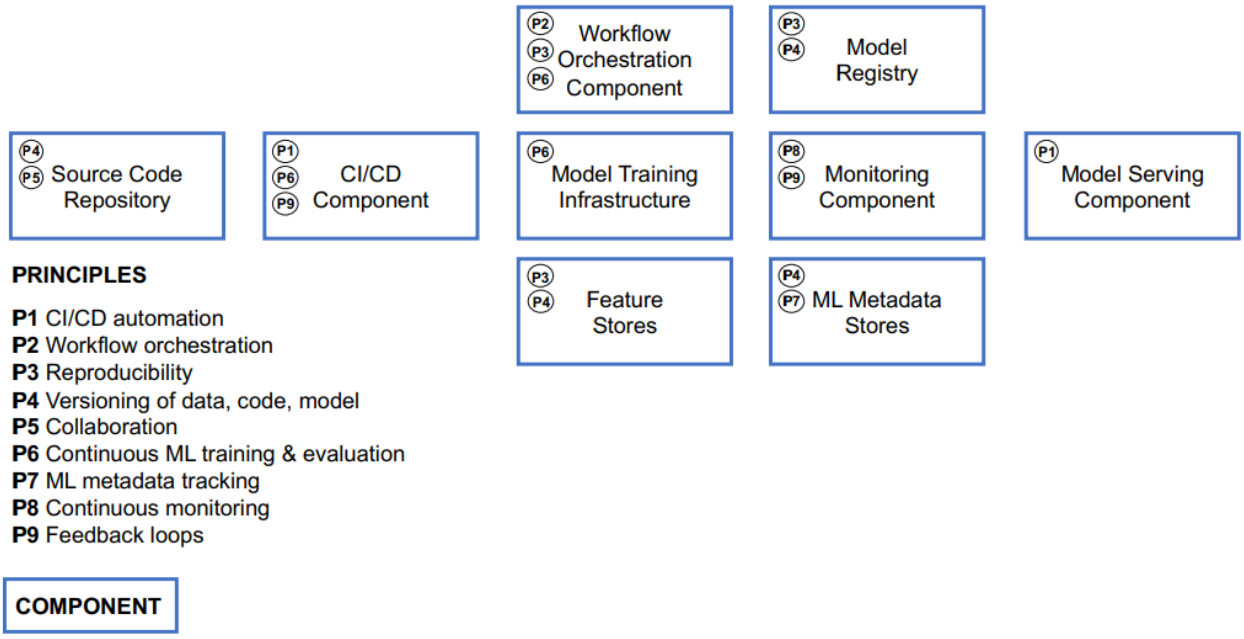

Figure 18. MLOps Principles within Technical Components. Source from [25]

Automation extends beyond basic scripting and includes sophisticated workflows that handle data ingestion, feature engineering, model training, and deployment with minimal human intervention. This level of automation accelerates the development process and ensures each step is repeatable and consistent. Collaboration is facilitated by tools and platforms that allow for shared access to data, code, and models, enabling team members to work together seamlessly. This collaborative environment helps in breaking down silos and encourages the sharing of insights and best practices. Continuous improvement is driven by metrics and feedback loops that inform teams about the performance of their models in production, enabling them to make data-driven decisions about when and how to update models. Governance involves establishing clear protocols for model validation, monitoring, and auditing, ensuring models operate within defined ethical and legal boundaries. By adhering to these principles, organizations can build a strong foundation for their MLOps practices, leading to more reliable and impactful ML solutions. The paragraph 'MLOps Frameworks and Tools' will list possible frameworks/tools that can support the MLOps principles described above.

4.3 MLOps Key Components

The key components of MLOps include infrastructure, data and model versioning, monitoring, orchestration, and governance. Infrastructure refers to computational resources, cloud services, and containerization technologies (like Docker [22] and Kubernetes [23]) that support scalable and efficient model training and deployment. These technologies enable seamless scaling, resource management, and deployment of models across different environments. Data and model versioning are essential for ensuring the correct versions of datasets and models are used in experiments and production, facilitating reproducibility, and providing a clear audit trail. Monitoring involves continuously tracking the performance, accuracy, and behavior of models in production to detect and address issues such as model drift, bias, or performance degradation. Orchestration tools, such as Argo Workflows [24], Kubeflow [25], and MLFlow [26], automate and manage the complex workflows involved in the ML lifecycle, from data ingestion and preprocessing to training, evaluation, and deployment. Governance encompasses policies and practices to ensure models are developed and deployed in a secure, compliant, and ethical manner. These components work together to create a comprehensive MLOps framework that enhances the efficiency, reliability, and scalability of machine learning operations.

The diagram architecture in Figure 19, adapted from the book "Introducing MLOps: How to Scale Machine Learning in the Enterprise" [67], illustrates a comprehensive range of MLOps functionalities and practices aimed at developing and maintaining Machine Learning models reliably and efficiently.

For each macro-step, represented by the four central circles and the diamond on the left, the figure includes a set of functionalities or requirements pertinent to the respective macro-step. Additionally, it indicates the roles of the individuals most involved in each macro-step.

Starting from the left, the very first phase is scoping, which involves defining the exact application of Machine Learning; determining what inputs "X" and outcomes "Y" are, and outlining all business aspects and improvements aimed to be achieved using an ML-based solution.

Once these questions are answered, the development of the ML model(s) can start. For development, it is necessary to first collect or acquire data for the algorithm. The data must be properly prepared and pre-processed to optimize model performance. This optimization involves improving the model's accuracy and other selected metrics, as well as enhancing computational efficiency, such as reducing training time and model size. . During the error analysis process, it may be necessary to update the model or return to the previous phase to gather more data.

The next phase is preparing the model for production deployment: an intermediate phase that prepares the runtime environment, conducts a risk assessment, and ensures specific quality standards for the machine learning model.

The following phase is the actual deployment of the model into production, which requires support for features such as scaling the model, containerization, and CI/CD of the entire process.

Finally, continuous monitoring of the model in execution involves online evaluation of the model; detection of data drift and model drift via scheduling of appropriate pipelines that detect changes in data distribution.

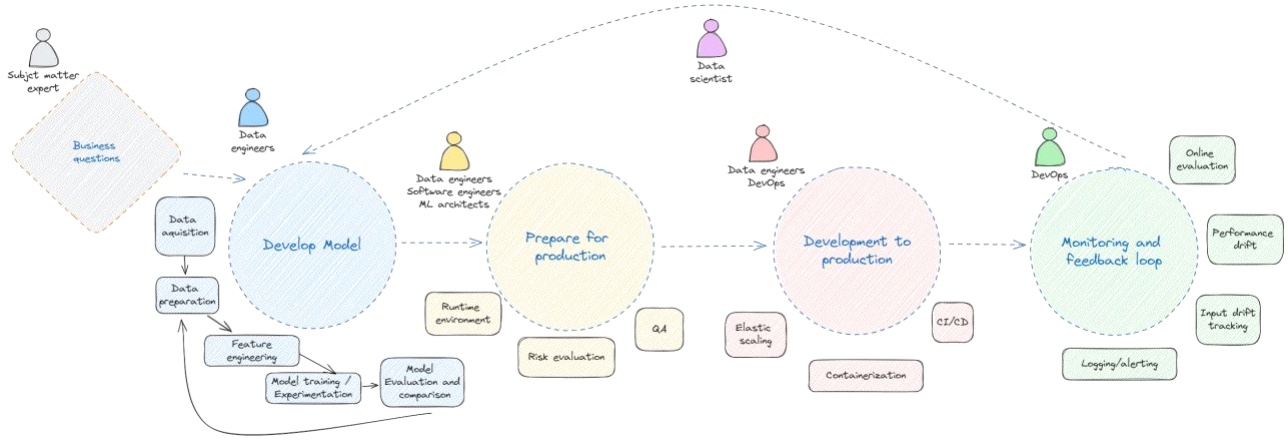

Figure 19. MLOps end-to-end Flow Architecture

Each of these components plays a critical role in the successful implementation of MLOps. Infrastructure provides the backbone for ML operations, ensuring that computational resources can be dynamically allocated and managed based on the needs of different stages of the ML lifecycle. Cloud platforms like AWS, Azure, and Google Cloud offer scalable resources and integrated services that simplify the deployment and management of ML models often through a 'pay-per-use' mode. However, there are many other open-source tools that can be exploited together to manage the life cycle of machine learning models also on-premises. Data and model versioning systems, such as MLFlow and Git, allow teams to track changes over time, ensuring that they can revert to previous versions if needed and maintain a history of all modifications. Monitoring tools like Prometheus [27] and Grafana [28] enable real-time tracking of model performance, alerting teams to any anomalies or issues that may arise. This continuous monitoring is crucial for maintaining model accuracy and reliability in dynamic production environments.

Orchestration tools, such as KubeFlow, Apache Airflow, and Argo Workflows manage the complex workflows involved in training and deploying ML models, automating tasks, and ensuring that dependencies are correctly handled. Governance frameworks ensure that all aspects of the ML lifecycle adhere to organizational policies and regulatory requirements, addressing concerns around data privacy, security, and ethical use of AI. Together, these components provide a solid foundation for MLOps, enabling organizations to scale their ML operations efficiently and effectively.

4.4 MLOps Frameworks and Tools

MLOps frameworks and tools play a critical role in streamlining the development, deployment, and maintenance of machine learning models. These frameworks provide essential functionalities such as continuous integration and deployment, workflow orchestration, reproducibility, and monitoring, which are vital for maintaining robust and scalable ML systems. In this paragraph, some frameworks and tools will be listed that can support the main principles of MLOps, including CI/CD automations, workflow orchestration, reproducibility, versioning of data, code, and models, collaboration, continuous ML training and evaluation, ML metadata and tracking, continuous monitoring, and feedback loops. Additionally, we will examine in the next paragraph two of the most widely used MLOps platforms and introduce a cutting-edge ENG research asset under development, known as ALIDA.

4.4.1 Continuous Integration (CI) / Continuous Deployment (CD) Automations

Continuous Integration (CI) and Continuous Deployment (CD) automations involve automatically integrating code changes, running tests, and deploying updates to production. In MLOps, this includes automating the testing of data transformations, model training, and validation processes to ensure that any changes in code or data do not break the pipeline and can be safely deployed to production environments. Examples of tools that can be used for these purposes include Jenkins [29], which provides extensive plugins for automating various stages of the CI/CD pipeline; GitHub [30] Actions, which offers native integration with GitHub repositories for CI/CD workflows; and GitLab [31] CI/CD, which offers built-in tools for continuous integration, delivery, and deployment. For MLOps-specific tasks, tools like KubeFlow, which provides machine learning toolkit for Kubernetes, and MLFlow, which facilitates the management of the machine learning lifecycle, are also commonly used. Additionally, Argo CD [32] is a declarative, GitOps continuous delivery tool for Kubernetes. GitOps (Git Operations) is a methodology for implementing Continuous Deployment for cloud-native applications by using Git as a single source of truth for declarative infrastructure and application definitions. With GitOps, the desired state of the entire system is versioned in Git, and automated processes ensure that the system's live state matches the state defined in Git. This approach enhances visibility, consistency, and reliability in managing Kubernetes clusters and applications.

4.4.2 Workflow Orchestration

Workflow orchestration manages the sequence and execution of different tasks and processes involved in an ML pipeline. This ensures that data preprocessing, model training, validation, and deployment steps are executed in the correct order and at the right times, also across different systems and environments. Tools like Apache Airflow, KubeFlow, or Argo Workflows are commonly used for this purpose.

As already described in the DataOps section, Apache Airflow is an open-source platform that allows you to programmatically author, schedule, and monitor workflows. It uses directed acyclic graphs (DAGs) to manage the execution order of tasks and can integrate with various data processing frameworks and services. This flexibility makes it suitable for orchestrating complex workflows that span multiple technologies and environments.

KubeFlow is a machine learning toolkit for Kubernetes, designed to simplify the deployment, scaling, and management of machine learning models. It provides a comprehensive suite of components for every stage of the ML lifecycle, including Pipelines, which offer a platform for building and deploying portable, scalable ML workflows based on Docker containers. Kubeflow Pipelines can handle a wide range of tasks, from data preprocessing to model training and serving, all within a Kubernetes cluster.

Argo Workflows is an open-source container-native workflow engine for orchestrating parallel jobs on Kubernetes. It is designed to run complex workflows, where each step in the workflow is executed as a container. Argo Workflows is particularly powerful for CI/CD pipelines and ML workflows due to its native integration with Kubernetes and its ability to handle large-scale, distributed workloads efficiently. It supports advanced features such as DAGs, parameter passing, artifact management, and more, making it an excellent choice for managing sophisticated workflows in a scalable and reliable manner.

These tools ensure that each step in the ML pipeline, from data ingestion to model deployment, is executed in a structured and automated manner. This orchestration not only helps in maintaining the sequence and dependencies of tasks but also provides robust monitoring and logging, error handling, and scalability, which are crucial for maintaining the reliability and efficiency of ML workflows in production environments.

4.4.3 Reproducibility

Reproducibility ensures that experiments and results can be reliably repeated. This is achieved by maintaining consistent environments, using version control for code, data, and configurations, and logging experiment parameters and outcomes. Reproducibility is essential for verifying results and ensuring the reliability of the models.

Maintaining consistent environments involves using tools like Docker and Kubernetes, which allow you to create and manage containerized environments that can be easily replicated across different systems. Docker containers encapsulate the software and its dependencies, ensuring that the code runs in the same environment every time, regardless of where it is deployed.

Version control for code, data, and configurations is crucial for tracking changes and ensuring that every aspect of an experiment can be traced and reproduced. Tools like Git are commonly used for versioning code, while data versioning can be managed using tools like DVC (Data Version Control) [33]. These tools allow teams to maintain a history of changes, compare different versions, and revert to previous states if needed.

Logging experiment parameters and outcomes is another key aspect of reproducibility. Tools like MLFlow help in tracking and managing experiment metadata, including hyperparameters, metrics, and artifacts generated during the experiment. This information is crucial for understanding the behavior of models and for comparing the performance of different experiments.

Furthermore, automated pipelines play a significant role in ensuring reproducibility. By automating the entire ML workflow, from data preprocessing to model deployment, tools like Apache Airflow, Kubeflow, and Argo Workflows help maintain consistency and reduce human error. These tools enable the definition of workflows as code, ensuring that the same steps are followed every time an experiment is run. These practices not only ensure that experiments can be reliably repeated but also enhance collaboration among team members, improve transparency, and build confidence in the results and models produced.

4.4.4 Versioning of Data, Code, and Model

Versioning involves keeping track of different versions of datasets, codebases, and models. This allows teams to trace back to specific versions used in experiments or production, facilitating debugging and compliance. Version control systems like Git for code, DVC (Data Version Control) for data, and model registries for models are typically used.

For code versioning, Git is the industry-standard tool. It allows developers to manage changes to their codebase, collaborate with others, and maintain a history of all changes. Git's branching and merging capabilities enable multiple team members to work on different features or fixes simultaneously, integrating their work seamlessly. This is critical for large-scale projects where maintaining a clean and organized codebase is essential.

Data versioning is handled by tools like DVC and Pachyderm [34]. DVC integrates with Git to manage large datasets and machine learning models, creating lightweight metafiles that track data versions. This enables data scientists to version their datasets in a similar way to code, ensuring consistency and reproducibility. Pachyderm, on the other hand, offers a data-centric approach to versioning, where each change to the data automatically triggers a new version, and complex data processing pipelines can be built and tracked. Both DVC and Pachyderm are flexible in the types of data they can handle as any type of file or data can be processed and stored in a file system, making them suitable for a wide range of data science and machine learning applications. In fact, DVC is file-agnostic and works by tracking files and directories in Git; Pachyderm is also file-agnostic and works by managing data through its version-controlled file system.

Model versioning is equally important in the machine learning lifecycle. Model registries like MLFlow Model Registry [68], Amazon SageMaker [35] Model Registry allow teams to store, version, and manage models in a centralized repository. These tools track the lineage of models, including the parameters, code, and data used to train them. This makes it easier to reproduce experiments, compare model performance, and deploy specific versions to production.

The benefits of robust versioning include:

1. **Traceability**: teams can trace the exact version of code, data, and models used at any point in time, which is invaluable for debugging issues or understanding the context of past experiments.
2. **Collaboration**: multiple team members can work on different aspects of a project simultaneously without interfering with each other's work. Version control systems manage these concurrent changes and merge them back into the main project seamlessly.
3. **Reproducibility**: by maintaining versions of datasets, codebases, and models, experiments can be reliably reproduced. This is critical for validating results and ensuring that models perform consistently when redeployed.
4. **Compliance**: many industries have regulations requiring the tracking of data lineage and model usage. Versioning helps meet these compliance requirements by providing a clear history of changes and usage.
5. **Rollback**: in case of issues with new changes, teams can quickly revert to previous stable versions, minimizing downtime and mitigating risks associated with deploying faulty updates.

Automated versioning systems can integrate with CI/CD pipelines to ensure that every change is tracked and tested before being deployed. This integration further enhances the reliability and stability of ML workflows.

4.4.5 Collaboration

Collaboration in MLOps refers to enabling seamless cooperation among data scientists, engineers, and other stakeholders. This includes sharing code, data, and models, providing feedback, and managing permissions and roles. Collaboration tools might include platforms like GitHub, GitLab, and shared workspaces such as Jupyter [36] notebooks or integrated development environments (IDEs).

4.4.6 Continuous ML Training & Evaluation

Continuous ML training and evaluation ensure that models are regularly retrained and assessed using new data. This helps in maintaining the model's performance over time as data distributions change. Automated retraining and evaluation pipelines can be set up to trigger based on new data availability or at scheduled intervals.

4.4.7 ML Metadata and Tracking

ML metadata and tracking involve logging all relevant information about the machine learning processes, including datasets, parameters, metrics, and environments. This helps in understanding the context of each experiment,

debugging, and improving models. Tools like MLFlow, Weights & Biases [37], and TensorBoard from TensorFlow [38] are commonly used for this purpose.

4.4.8 Continuous Monitoring

Continuous monitoring involves keeping track of model performance and data quality in production. This ensures that models are functioning as expected and helps in detecting issues like data drift or model degradation early. Monitoring tools can include Prometheus, Grafana, or specialized ML monitoring platforms.

4.4.9 Feedback Loops

Feedback loops in MLOps involve incorporating user and system feedback into the model improvement cycle. This can include user interactions, performance metrics, and error reports. Effective feedback loops help in iteratively improving model performance and adapting to changing conditions. Automated mechanisms for collecting and integrating feedback are essential for maintaining model relevance and accuracy.

4.5 MLOps Platforms

This paragraph will describe some MLOps platforms such as MLFlow and KubeFlow, and an ENG research asset under development that applies and supports MLOps methodologies.

4.5.1 MLFlow

MLFlow is an open-source platform designed to streamline the machine learning lifecycle, including experimentation, reproducibility, and deployment. Developed by Databricks, it is highly adaptable and can be integrated with a variety of ML libraries and tools.

Key Components of MLFlow:

1. MLFlow Tracking: this component allows users to log and query experiments, including code, data, configuration, and results. It helps in organizing and retrieving experiment data easily.
2. MLFlow Projects: MLFlow Projects provide a standardized format for packaging data science code. A project is simply a directory with an MLProject file that defines the project and how to run it.
3. MLFlow Models: this component manages and deploys models from various ML frameworks. Models can be stored in a central repository and deployed to multiple environments.
4. MLFlow Registry: the Model Registry is a centralized store to manage the lifecycle of MLFlow models, including versioning, staging, and annotation.

The diagram in Figure 20 illustrates the architecture of MLFlow, showing how the different components interact to manage the ML lifecycle. Experiments are tracked and logged, projects are packaged for reproducibility, models are managed and deployed, and the registry oversees the entire lifecycle.

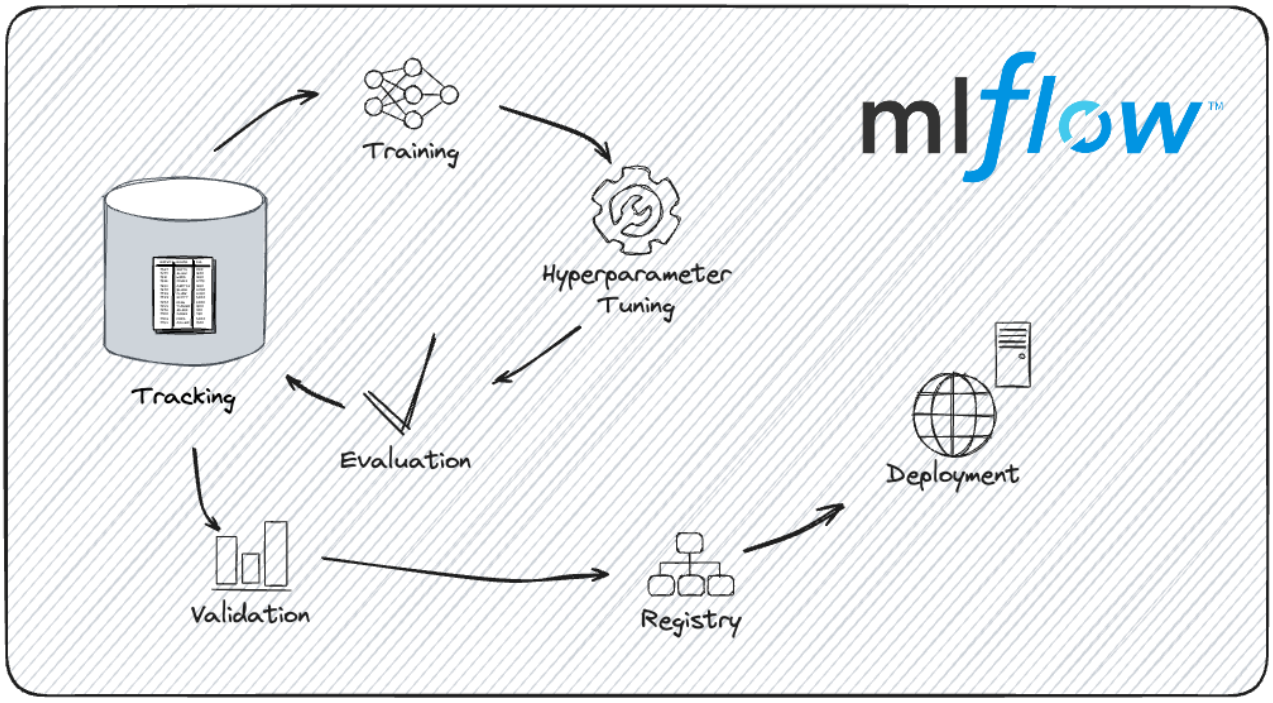

Figure 20. Model Development Lifecycle with MLFlow. Source from https://mlflow.org

Some benefits of using MLFlow:

- Ease of use: easy to set up and use with straightforward APIs.
- Integration: compatible with many ML libraries such as TensorFlow, PyTorch, and Scikit-Learn.
- Flexibility: supports a wide range of deployment options, including local, cloud, and edge environments.

4.5.2 KubeFlow

KubeFlow is an open-source platform that aims to simplify the deployment, orchestration, and management of machine learning workflows on Kubernetes. It leverages Kubernetes' scalability and flexibility to manage complex ML workflows.

Key Components of KubeFlow:

1. Kubeflow Pipelines: a platform for building and deploying portable, scalable ML workflows based on Docker containers. Pipelines provide a way to orchestrate ML tasks in a reproducible manner.
2. KFServing: a serverless framework for serving machine learning models on Kubernetes, supporting multiple frameworks such as TensorFlow, XGBoost, and PyTorch.
3. Katib: a Kubernetes-native system for hyperparameter tuning and neural architecture search, enabling automated model optimization.
4. Notebooks: Kubeflow integrates with Jupyter Notebooks, providing a collaborative environment for data scientists to develop and test their models.

The diagram in Figure 21 illustrates the architecture of KubeFlow, showcasing how the different components interact within a Kubernetes cluster. Pipelines manage the orchestration of ML workflows, KFServing handles model serving, Katib optimizes models through hyperparameter tuning, and Jupyter Notebooks provide a development environment.

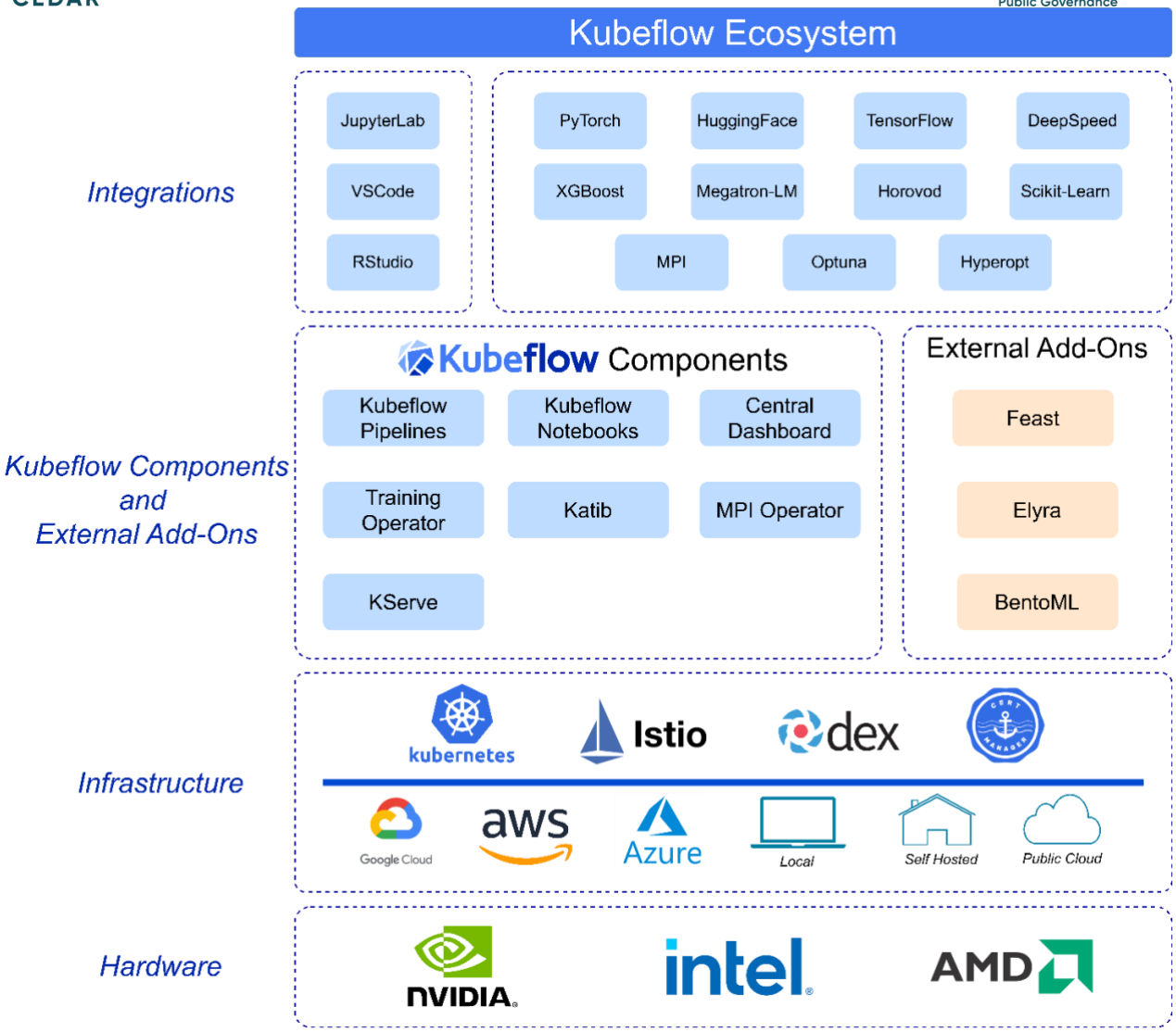

Figure 21. KubeFlow Concept Architecture. Source from https://kubeflow.org

Some benefits of using KubeFlow:

- Scalability: leverages Kubernetes' scalability to handle large and complex workflows.
- Modularity: allows for modular and reusable components, making it easier to manage and extend.
- Comprehensive: provides end-to-end support for the entire ML lifecycle, from data preparation to model serving.

4.5.3 ALIDA: an ENG cutting-edge research platform

As the field of machine learning continues to grow, innovative platforms are being developed to enhance MLOps capabilities. One of these platforms is ALIDA, a research asset currently under development by ENGINEERING (https://eng.it/en/case-studies/alida-per-migliorare-la-flessibilita-reattivita-del-business). ALIDA is a DSML (Data Science and Machine Learning) platform that utilizes a micro-service architecture and no/low-code tools to streamline the creation and execution of machine learning pipelines.

ALIDA stands out by offering a no-code interface that leverages frameworks like Argo Workflows for the workflows' orchestration, enabling users to graphically design and deploy ML pipelines with ease. This user-friendly approach

aims to make advanced ML workflows accessible to a broader audience, including those without extensive coding expertise.

For the CEDAR project, the ALIDA platform will be extended integrating other MLOps tools such as MLFlow for experiment tracking and additional solutions for model serving and monitoring. This integration will support comprehensive MLOps activities, making ALIDA a tool for accelerating the key phases of the machine learning lifecycle. In the CEDAR project, ALIDA will enhance MLOps practices and significantly speed up the development, deployment, and management of ML models. Through its advanced features and focus on usability, ALIDA aims to provide an efficient and effective platform for supporting all aspects of the ML lifecycle. Using the ALIDA platform, users can design their own stream/batch workflows by choosing from the ALIDA catalog Big Data Analytics (BDA) services and the set of big data to process, run and monitor the execution. The resulting Big Data Analytics applications can be deployed and installed in another target infrastructure with the support of a package manager that simplifies deployment within the target cluster.

ALIDA, as a cloud-native platform, can scale computing and storage resources thanks to a pipeline orchestration engine that leverages the capabilities of Kubernetes for cloud resource management. ALIDA provides an extensible catalogue of BDA services (the building blocks of the BDA Application) which covers all phases, from ingestion to preparation, to ML analysis and for data publishing. Figure 22 shows an overview of ALIDA and outlines its features, including the possibility to process and do ingestion of data from different sources; the integrations and registrations of new BDA services also developed by third parties; the possibility to put together these BDA services through a web-based designer to compose a BDA Application formed by assets (datasets and/or models) and BDA services; and supporting both batch and streaming workflow design and execution.

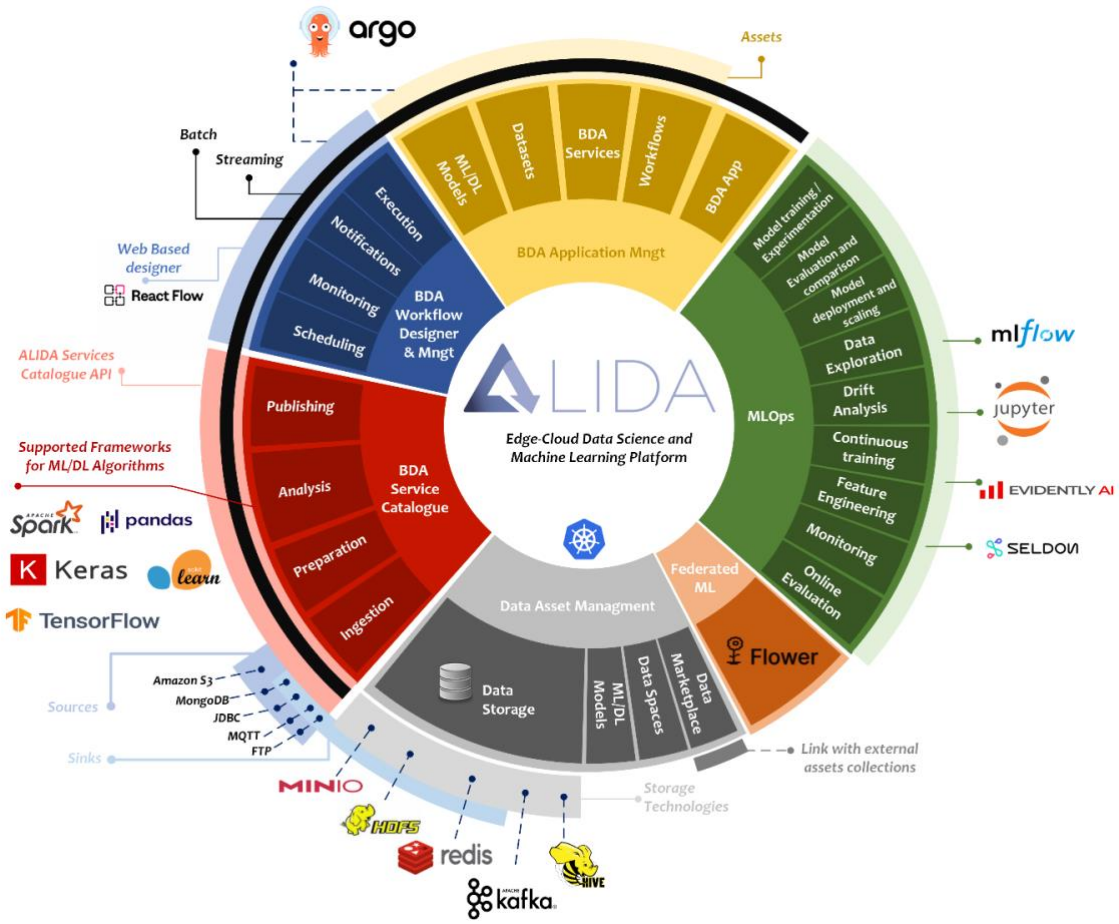

Figure 22. ALIDA General Overview

ALIDA supports the composition, deployment, and execution of BDA applications through a graphical user interface. Each BDA application is composed of one or multiple BDA services. Each BDA service is a containerized OCI-compliant micro-service application. ALIDA platform is based on and/or uses different open-source technologies, among which:

- Kubernetes: as resource orchestrator.
- Argo Workflows: an open-source container-native workflow engine for orchestrating parallel jobs on Kubernetes.
- Apache Kafka [39]: an open-source distributed event streaming platform.
- React Flow [40]: a component used to design pipelines of BDA services.

Figure 23 shows the ALIDA architecture. The main components of ALIDA shown in the architecture are:

- ALIDA GUI (Graphical User Interface): the graphical user interface of ALIDA, based on React [41].
- ALIDA Platform: the main ALIDA back-end service, responsible for managing all requests coming from the GUI and other components.
- BDA Application Catalogue: contains all operations (creation, reading, updating, deletion) related to the objects managed in ALIDA: datasets and models metadata, BDA services, BDA applications, and more.
- Asset Provider: component of ALIDA responsible for parsing the BDA application pipeline generated through the ALIDA graphical designer and providing assets such as exporting the BDA application and the generated model.
- K8S Client: component for executing/stopping BDA applications workflows and sending BDA application events that are executed in ALIDA, interacting with Kubernetes and Argo Workflow APIs.
- Status Manager: based on incoming status events, manages and updates the states of the BDA applications by interfacing with the BDA Catalogue.
- Notification Manager: the notification manager is the core of the notification system, it subscribes to Kafka topics inside the platform and censuses the notifications in a database; it also handles the management of user preferences and based on these performs triggers and sends notifications to the clients; notifications, which may contain simple logs, images, or even more, are sent by BDA-services executed within the ALIDA platform using an extension of the ALIDA libraries.

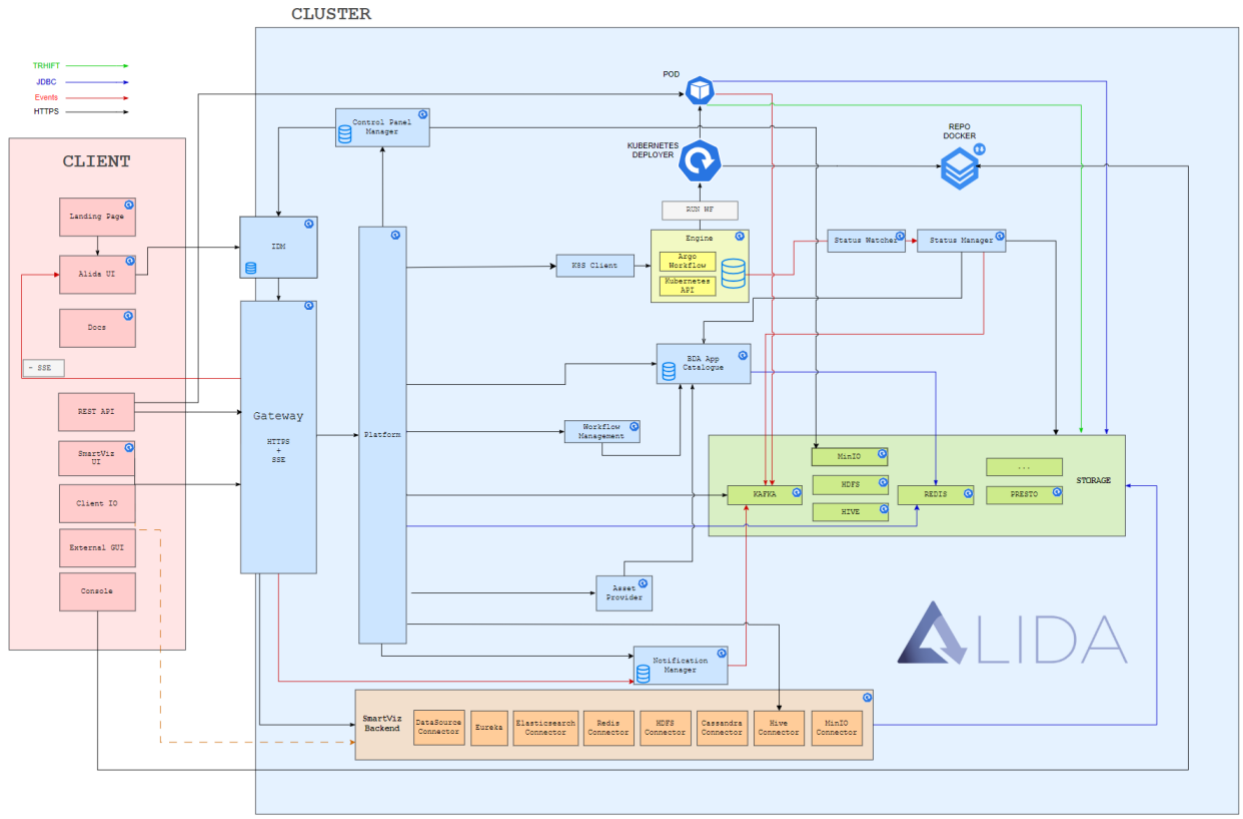

Figure 23. ALIDA Architecture

4.5.4 MLOps and CEDAR project

The CEDAR project will leverage and extend the features provided by the ALIDA platform, including the design and execution of ML analytics pipelines. The functionalities of the ALIDA platform will be expanded to support the deployment and monitoring of trained ML models. For deployment, open-source frameworks such as Seldon Core [42] will be considered. To standardize the training process, the use of MLFlow will be promoted, particularly for partners in Work Package 4 (WP4) responsible for ML model creation.

The choice of Seldon Core is motivated by its powerful capabilities for deploying, scaling, and monitoring machine learning models in Kubernetes environments. This allows for efficient management of model serving in production. MLFlow, with its comprehensive support for the entire machine learning lifecycle, will standardize how models are trained, tracked, and packaged, ensuring consistency and reproducibility across different teams and partners.

By extending the ALIDA platform, we aim to enhance the deployment and monitoring of trained ML models, including detecting model degradation or changes in data distribution. This will involve setting up notification systems and pipelines for the automatic retraining of models, ensuring continuous improvement and adaptation to new data. This comprehensive approach ensures that the models remain effective and reliable over time, maintaining their performance in dynamic data environments.

In conclusion, the integration of these advanced MLOps practices within the CEDAR project underscores our commitment to creating a robust, scalable, and efficient machine learning infrastructure. This will not only enhance model deployment and monitoring but also ensure that our models continue to deliver high performance and value in a rapidly evolving data landscape.

5 Integration with CEDS

5.1 Overview

Integration with CEDS is the main goal of Task 3.3. More specifically, this task aims to develop dataspace connectors that align with key European initiatives like IDSA, GAIA-X, EOSC, etc, facilitating continuous data exchange and interoperability among these initiatives, to create a secure, reliable, and integrated European data network. The first step of action includes an in-depth analysis of each initiative's technical specifications, focusing on identifying supported data formats and protocols. Utilizing the insights from this analysis, the second step is to design suitable APIs, while also managing the alignment of data. Task 3.4 will contribute knowledge on necessary security protocols, in order to ensure that the connectors bridge various systems and platforms not only efficiently, but also with enhanced security. Then, the connectors will undergo extensive testing to confirm their capability to handle data transfers smoothly and securely. The goal of this task is to support a solid and interoperable European data ecosystem that enhances data sharing and collaboration, while maintaining high data security and privacy standards.

As Task 3.3 progresses, it will establish a fully operational data space within the CEDAR platform. This will bring CEDAR a step closer to the vision of being compliant with the Common European Data Spaces (CEDS). The concept of CEDS is part of the European strategy to enhance data sharing across different sectors and borders within the European Union (EU). These data spaces are intended to create a single market for data, ensuring that data can flow freely across the EU, while still being subject to high standards of data protection, privacy, and security. Therefore, the establishment of CEDS is crucial for Europe to operate in a unified manner that reflects core European values such as self-determination, privacy, transparency, security, and fair competition.

Based on the DoA and according to the European Strategy for Data, nine specific CEDS have been designated to enhance research and innovation across various sectors. These include Health, exemplified by the MUSKETEER and KRAKEN projects; Industry & Manufacturing, highlighted by the i3-Market initiative; Agriculture & Food Supply Chain, represented by TheFSM project; and sectors like Culture and Mobility, with initiatives such as MobiSpaces, DataPorts, and DataVaults. Additionally, efforts in Energy & Green Deal are advanced through projects like SmashHit and BD4NRG, while Security is addressed by TENSOR and LAGO, and Public Administration & Media are supported by initiatives like STARLIGHT and LAGO. CEDAR envisions the existence of a cross-border, cross-sectoral data sharing space that allows platforms to handle a combination of proprietary, personal, and open public data, while addressing existing technical and business challenges. Thus, CEDAR's goal of being compatible with other CEDS initiatives is one step towards this vision. However, to proceed to such an important step, it is crucial to begin with the analysis and collection of appropriate connectors that will be used in CEDAR platform's data space.

The objective of this first step is to identify the most mature Connectors by the second quarter of 2024, focusing on those that are well-developed and suitable for further distinction. These selected connectors will serve as the foundational elements for analyzing the technical specifications of compatible initiatives. The approach involves selecting a Dataspace Connector that not only meets the project's specific requirements but also aligns with multiple Dataspace Initiatives. This strategy guarantees that the chosen connectors are both advanced in their development and compliant with broader initiative standards, enabling effective and efficient data integration across diverse platforms.

5.2 Connectors Research

5.2.1 IDS Data Connector Report

The Data Connector Report [43], published regularly by IDSA, is designed to explain data connectors. This current research is based on Report 89 (No13) of March 2024. In summary, the report covers:

- **Explanation of Data Connectors:** it details what data connectors are and why they are crucial in data spaces.

- **Types of Connectors**: the report categorizes data connectors into four types: data connector frameworks, open-source generic solutions, proprietary generic solutions, and off-the-shelf data connectors or those integrated in data-related products.
- **Interoperability Requirements**: it outlines the requirements for making data connectors work together smoothly, like adhering to standards, clear specifications, and promoting semantic interoperability using specific vocabularies such as the Data Catalog Vocabulary (DCAT).
- **Visibility of Implementations**: the report showcases existing implementations, providing information on their license type, maturity, and usage cases, and tracks their evolution.
- **Learning and Enabling Interoperability**: it aims to be a learning hub for data sharing ecosystems, discussing other approaches that help in data-driven business ecosystems and promoting future alignment with IDS.
- **Additional Technologies**: additional technologies like the Gaia-X trust framework, iShare, and SOLID are listed, which aid in trustworthy data sharing.

The report encourages feedback and updates about listed connectors or suggestions for new technologies. It focuses specifically on connectors, not on IDS use cases or deployment scenarios. In addition, the report provides comprehensive details on various data connectors with different maturity levels, as measured by Technology Readiness Levels (TRLs). These TRLs could be categorized as follows:

- **Higher TRLs (7-9)**: these connectors are closer to or are already in the deployment phase and are being used in real-world applications. **The focus of CEDAR research will be on connectors of this TRL group**, since they have proven to be effective and are already exploring optimization, integration with other technologies, or specific case studies.
- **Mid-range TRLs (4-6)**: these connectors are in the developmental or testing stages and have not been fully commercialized yet. This area could be ideal for research, if the goal would be to contribute to the evolution of connectors, solve emerging problems, and enhance existing capabilities before they reach the market.
- **Lower TRLs (1-3)**: these are in the early stages of concept and design. Research here could be interesting for similar reasons to the Mid-range TRL connectors; however such an approach seems to be out of CEDAR scope.

5.2.2 Connector Selection Criteria

The connectors included in the IDS Data Connector Report will be filtered based on three factors. The first factor is the TRL level. Connectors below TRL 7 will not be considered, since the final dataspace implemented within CEDAR will have to be TRL 7. As mentioned above, the High TRL connectors are the ones listed with Technology Readiness Levels between 7 and 9. The only exception will be the TRUE Connector, due to its relatively high adoption rate. The second factor is the open-source nature of each connector. Although both closed-source and open-source connectors will be analyzed, the final selection will be made from the open-source connectors' list. The third factor will be whether each connector has a Minimum Viable Dataspace (MVD) implemented or not. An MVD implementation is crucial as it not only demonstrates the ease of deployment for a specific connector but also validates (or refutes) the Technology Readiness Level (TRL) claimed by the connector's vendor. This process ensures that the connector's capabilities are accurately represented and reliable in practical applications. One last filter that will be examined is whether each connector is IDS Certified or not. However, this filter will not affect the final connector selection.

5.2.3 Open-Source High TRL Connectors

5.2.3.1 Boot-X Connector

The Boot-X Connector [44], maintained by Huawei at their Munich Research Center, is **designed as a Gaia-X / IDSA compliance-ready**, cloud-based data space implementation, ***focusing on cross-border data exchange, particularly***

between Chinese and European industries. This connector, which is compatible with the Eclipse Data Space connector, includes enhanced features such as local data policy, Self-Sovereign identity federation, and compliance monitoring. It is categorized under multiple types: a data connector framework, a generic solutions software, and an off-the-shelf solution provided as a service. With a maturity level of **TRL 7**, the Boot-X Connector is currently live on the IDSA Radar. It is platform-agnostic in terms of portability and **has an open-source license** that is yet to be determined. Although **it lacks IDS Certification**, it has been implemented in the domain-agnostic Boot-X platform, which supports cross-border data exchange and interoperates with Huawei Cloud's EDS (Exchange Data Service) in China.

For deployment, the Boot-X Connector offers options such as on-premises, cloud, and specifically Huawei Cloud environments. The service level includes both connector as a service and platform as a service. The connector features robust access and usage control capabilities, including OAuth and Basic Auth for access control and extensive usage control policies that cover aspects from data consumption to data deletion. Communication is facilitated through the Dataspace protocol (HTTPS) and includes both in-band and out-of-band transfer protocols. It also features a comprehensive graphical user interface for users, management, and administration, and supports identity management with centralized (X.509), decentralized (did:web), and SSI, enhanced by a credential bridge for SSI and OIDC. Despite not supporting the IDS Information Model, the Boot-X Connector integrates with the EDC Catalogue and does not currently integrate with a Clearing House. For more detailed information and updates, the Boot-X website provides further resources.

5.2.3.2 ECI Gatewise IDS Connector powered by TNO

The ECI Gatewise IDS Connector [45], **co-maintained by ECI Software Solutions and TNO**, is a cloud-based connector ***designed for digital exchange of supply chain-related messages*** between companies affiliated with different SCSN Service Providers. It is categorized as a data connector framework, a generic open-source solution, and an off-the-shelf solution provided as a service. This connector is at a high maturity level with a **TRL of 9**, indicating it is fully operational ("live"). While it supports specific platforms (not platform agnostic), its licensing is **open-source for the components developed by TNO**, though **developments by ECI Software Solutions are not open-source**.

The ECI Gatewise IDS Connector is **not IDS certified**. It can be used in the manufacturing sector for ***exchanging order-related data between manufacturing companies and wholesalers***. It offers deployment solely in cloud environments. The service level includes both connector as a service and self-service options. For security, it employs OAuth and API key mechanisms for access control. It also supports usage control, but details on specific usage control policies are not provided in the report. The communication is managed through the IDS-REST protocol. Additionally, it does not provide a graphical user interface, support the IDS Information Model, or include integration with Catalogue/Meta Data Brokers and Clearing Houses. Further information can be found on TNO's website on IDS technology.

5.2.3.3 Eclipse Dataspace Components (EDC) – Framework

The Eclipse Dataspace Components (EDC) – Framework [46], **managed by the Committer Group in the Eclipse Foundation**, is designed to be ***highly adaptable for any setup***, including on-premises, various cloud vendors, hybrid environments, or even individual end-user machines. It is a data connector framework specifically developed to ***allow customization and scalable data space construction***. Notably, the EDC separates control and data planes, facilitating a modular approach to building data spaces, and supports contract negotiation and policy handling through common interfaces. The framework's maturity is rated at **TRL 8-9**, highlighting its advanced stage of development and readiness for integration into live environments. It requires a Java-based environment to operate and is available under an **open-source Apache 2.0 license**. However, **it has not received IDS Certification**.

EDC is used in several data space projects, such as Catena-X, Eona-X, Mobility Data Space-MDS, and Omega-X, among others. It does not specify deployment options, suggesting its flexibility, and the service level is determined

by the effort of the open-source community. In terms of security and operational features, EDC does not include built-in access or usage control, which are subject to data planes defined by specific implementations. It does not provide a graphical user interface or support for identity management. Additionally, the EDC does not integrate with Catalogue/Meta Data Brokers (yet) or Clearing Houses and does not support the IDS Information Model or any vocabulary. For more detailed information or to access the source code, resources are available at the EDC homepage and the source code repository provided by the Eclipse Foundation.

5.2.3.4 EONA-X EDC Connector

The EONA-X EDC Connector[4], **maintained by Amadeus**, is designed ***to connect participants in the mobility, transport, and tourism sectors***. It is an off-the-shelf solution that can be provided as a service or directly used when integrated into data-related products. With a maturity level indicating it is **production-grade**, the EONA-X EDC Connector is therefore registered as ready for deployment. The connector is characterized by its agnostic portability, meaning it can be used across different platforms without dependency on a specific environment. It is **open-source under the Apache-2.0 license** but **has not yet received IDS Certification**. In terms of deployment, the EONA-X EDC Connector offers flexible options including on-premises and cloud environments, categorized under both connector as a service and self-service models. It supports OAuth for access control but does not include mechanisms for usage control.

Communication within the EONA-X system is facilitated through the Dataspace protocol (HTTPS) using in-band transfer protocols with not determined bindings. While it does not provide a graphical user interface, the connector supports decentralized identity management (did:web) but does not incorporate the IDS Information Model or any vocabulary. The EONA-X EDC Connector is integrated with a Catalogue/Meta Data Broker, specifically an EDC Federated Catalog, but does not connect with a Clearing House. This integration emphasizes its capability to participate actively in data ecosystems, particularly in scenarios involving significant events such as the Paris 2024 Olympics, where it will support data exchanges necessary for managing delegations during their arrival and departure. For further details and to explore additional functionalities, information is available on the EONA-X website.

5.2.3.5 IIOC IoT Connector (Intel IONOS Orbiter Connector)

The IIOC IoT Connector, **maintained by truzzt**, is an IoT version of the IDS Connector, compatible with the Eclipse Data Space Connector (EDC), and ***designed for minimal resource consumption, suitable for sensors and small devices***. It is developed in Rust & C, enhancing its efficiency for IoT applications. The connector is categorized as both a **generic open-source solution (with Apache 2.0 license)** and an off-the-shelf solution provided as a service. It holds a maturity level that signifies it is **already live and actively used within IDSA Base Camp**. The IIOC IoT Connector supports deployment in cloud environments and directly on IoT/CPS/OT devices, providing versatile implementation options. The connector **is not IDS Certified**.

Service levels offered include connector as a service, platform as a service, and self-service. It utilizes API keys for access control but does not support usage control, focusing instead on straightforward, secure connectivity. Communication protocols include IDS Multipart and the Dataspace protocol (HTTPS), with out-of-band data transfer utilizing protocol bindings. This connector also offers a graphical user interface for users, management, and administration. For identity management, it supports decentralized SSI. While it uses the IDS Information Model (Version 4.x), it does not support any vocabulary provision. Integration capabilities include connections with both Catalogue/Meta Data Brokers and Clearing Houses, specifically using the Base Camp Federated Catalogue and Base Camp Clearing House, facilitating broad interoperability and secure data exchange within diverse IoT ecosystems.

5.2.3.6 OneNet Connector

The OneNet Connector [47], **managed by Engineering Ingegneria Informatica S.p.a. and EUROPEAN DYNAMICS Luxembourg S.A.**, is designed to ***enable a European Energy Data Space, merging IDS principles with the FIWARE***

ecosystem's benefits for secure and decentralized data exchange. It is based on the TRUE Connector. This integration ensures seamless compatibility with existing platforms via APIs and includes full integration with the FIWARE Context Broker (NGSI-LD version). It offers a comprehensive graphical user interface for various services including KPIs, data exchange timelines, and cross-platform services catalogs. The connector is at a **TRL of 7** as of August 2023, demonstrating its operational readiness in multiple European environments. It is categorized as a **generic open-source** solution and is expected to be licensed under GPL v3 or similar upon the project's end.

OneNet Connector supports robust access control mechanisms using Basic Auth and API key, alongside extensive usage control policies. It utilizes the Dataspace protocol (HTTPS) for secure communication and supports in-band transfer protocols. While the connector features centralized identity management (X.509), **it is not IDS certified**, indicating it has not undergone or completed the specific compliance assessment with the IDS standards. Furthermore, the OneNet Connector does not integrate with a Catalogue/Meta Data Broker but is fully integrated with a Fraunhofer Clearing House. It does not support the IDS Information Model but is compatible with various vocabularies relevant to its operational domain. Additional information and features can be explored through the OneNet project website.

5.2.3.7 sovity Open-Source EDC Connector

The sovity Open-Source EDC Connector [48] [49], **managed by sovity GmbH**, is designed as a versatile and easy-to-use solution for ***data sharing among participants in data spaces***. It is based on the Eclipse Dataspace Components (EDC) framework (it is an extension of the EDC Connector) and is characterized by its ready-to-use, **open-source** nature, which is aimed at ***enhancing usability with features such as usage control***. The connector is available under an Apache 2.0 license and has reached a maturity level of **TRL 9**, indicating it is currently used in production environments. This connector is platform agnostic, allowing for deployment on-premises, in the cloud, or other environments, and is classified as a self-hosted service. It supports access control through mechanisms like Basic Auth and API key, with usage control policies including Connector Restriction and Time Interval implemented to manage data interactions securely and efficiently.

The sovity Open-Source EDC Connector utilizes the Dataspace Protocol (HTTPS) for communication and supports out-of-band data transfer protocols. It features a graphical user interface that facilitates ease of use for users, management, and administration. The connector also incorporates centralized DAPS (X.509) and mock IAM for identity management. However, the connector does not support the IDS Information Model but does integrate with Catalogue/Meta Data Brokers and Clearing Houses, enhancing its interoperability within the data space ecosystem. Last but not least, **the connector is not IDS certified**.

5.2.3.8 TNO Security Gateway (TSG)

The TNO Security Gateway (TSG) [50], **maintained by TNO**, is a multipurpose connector that facilitates robust data exchanges. This connector is an **open-source solution** and is platform agnostic, allowing it to be deployed flexibly in cloud environments. It has achieved a maturity level of **TRL 8**, indicating its readiness for operational use.

TSG is one of the connectors to be **IDS certified**, having already completed the Concept Review stage. It supports OAuth and API key methods for access control and includes usage control policies that manage aspects such as security level, time interval, location, and the number of usages. The connector's communication protocol is IDS Multipart, ensuring compatibility with various data handling and transfer methods. The TSG features a graphical user interface that caters to users, management, and administration, facilitating ease of use and comprehensive system control. It supports centralized identity management with X.509 certificates and integrates with the MetaData Broker Open Core, although it does not currently integrate with a Clearing House. For adoption, ***TSG is actively used in the Smart Connected Supplier Network (SCSN)*** [51] ***and various European and Dutch projects, showcasing its applicability and effectiveness in production settings.***

5.2.3.9 Trusted Connector

The Trusted Connector [52], maintained by **Fraunhofer AISEC**, is designed to ***enable enforceable usage control through integration with a Trusted Execution Environment and Remote Attestation***. It is a generic solution software, directly integrable into IT landscapes, often acting as proxies or gateways to companies' IT services. The connector is at a maturity level of approximately **TRL 7** and is available under an **open-source Apache 2.0 license**. Although it is moving towards IDS certification (it is an IDS Ready Component), it is **not yet IDS certified**. The Trusted Connector supports deployment across various environments, including edge, on-premises, and cloud setups, categorized under a platform as a service model.

It incorporates both OAuth and Basic Auth for access control and has defined usage control policies focused on applications within the connector and time intervals. In terms of communication, it utilizes the IDS Multipart and IDS protocol (IDSCP) without specific transfer protocols. The connector provides a graphical user interface designed for administration tasks. For identity management, it supports centralized systems (X.509). While it does not support the IDS Information Model or any specific vocabulary, it is integrated with the Fraunhofer Meta Data Broker and includes a connection to the AISEC-provided Clearing House.

5.2.4 Closed-Source High TRL Connectors

5.2.4.1 AI.SOV Connector

The AI.SOV Connector [53], maintained by Cefriel, is **built on the Fraunhofer open connector** enhanced with a resources catalogue from Cefriel, utilizing the KCong asset. It is designed as ***a user-friendly data exchange platform for the supply chain domain***, adhering to IDS data sovereignty principles. This connector is **closed source** (but extendable), classified as proprietary generic software and is at a maturity level of **TRL 7**, indicating its active use in data exchange domains. It is agnostic in terms of portability, meaning it can be deployed across various platforms, and features a closed source but extendable license.

The connector **has not received IDS Certification**. It has been implemented in several practical scenarios, including usage by Whirlpool and Sonae Arauco for data sharing within their supply chains. It has also been adopted by an automotive company to gather data from loggers in manufacturing plants. Deployment options include edge computing, on-premises, and cloud environments, with the service level categorized as "platform as a service." The connector supports OAuth and Basic Auth for access control and includes comprehensive usage control policies. *Communication is managed through the IDS protocol (IDSCP)*, and it features a graphical user interface for users, management, and administration purposes. However, it does not support any vocabulary or integration with Catalogue/Meta Data Brokers and Clearing Houses. For more information, one can refer to the AI.SOV GitLab page.

5.2.4.2 GDSO Connector - Tyre Information Service

The GDSO Connector - Tyre Information Service [54] is **maintained by the Global Data Service Organisation for tires and automotive components**. It facilitates ***communication using a REST API with a vocabulary consisting of standardized datasets for all GDSO Members***. The meta data broker acts as a resolver, providing information about endpoints offered by GDSO Members. The connector is a generic solution software that is **partially open source (and therefore listed as closed source here)**, with a maturity level at **TRL 9**, indicating it is fully operational and live. The connector is platform-agnostic and offers deployment options across edge, on-premises, cloud, and IoT/CPS/OT devices. It is classified as a platform as a service in terms of service level***. The connector is adopted by different stakeholders along the tyie value chain***: tire manufacturers, vehicle manufacturers, distributors and others. It **does not have IDS Certification**.

It employs OAuth for access control and supports extensive usage control policies tailored for B2B contract negotiations among GDSO Members, although the management and finalization of these contracts occur outside the GDSO framework. The communication protocol used is REST API, and it operates in-band with determined

protocol bindings for transfer protocols. The GDSO Connector does not feature a graphical user interface and supports centralized identity management based on AWS Cognito. It does not support the IDS Information Model but integrates with a Catalogue/Meta Data Broker, specifically a resolver that manages information about data endpoints offered by GDSO Members. There is no integration with a Clearing House. For further information, the GDSO website provides additional documentation and technical details.

5.2.4.3 Kharon IDS Connector powered by the Dataspace Connector

The Kharon IDS Connector [55], **maintained by HOLONIX SRL**, is embedded within a comprehensive IoT asset management solution known as **Kharon**. This integration facilitates the ***management of IoT data and augmented intelligence results through IDS***, allowing companies ***to expand their IoT networks securely while maintaining data sovereignty and industrial confidentiality***. This connector is categorized as a **proprietary generic solution (and close source code)** and an off-the-shelf solution provided as a service, with a maturity level of **TRL 7**, indicating its readiness for practical application in real-world settings. It **does not have an IDS Certification**.

Kharon IDS Connector employs OAuth for access control and supports usage control within the Kharon platform, although this is not yet implemented directly in the connector itself. Communication is handled through the IDS-REST protocol, and while the connector features a graphical user interface for management, it does not yet support decentralized identity management integration, which is under development (for Kharon, identity management is centralized). The connector does not support the IDS Information Model or any vocabulary, and it does not integrate with Catalogue/Meta Data Brokers or Clearing Houses. For more information, resources are available on the Holonix website and the Dat4Zero project website.

5.2.4.4 sovity CaaS (Connector-as-a-Service)

The sovity CaaS, **provided by sovity GmbH**, is an off-the-shelf solution designed for ***easy data sharing between participants in data spaces like MDS and Catena-X***. This fully managed, ready-to-use connector is ***based on the EDC*** (Eclipse Dataspace Components) framework and is particularly noted for its ***enhanced usability features which include usage control***. It operates under a **closed-source license** and is classified at a **TRL of 9**, indicating that it is already used in production environments. The sovity CaaS is platform agnostic, allowing deployment on-premises or in the cloud among other environments, and offers a service level that is Connector-as-a-Service.

It supports a variety of access controls including OAuth 2.0, Basic Auth, and API Key, and it provides specific usage control policies like Connector Restriction and Time Interval. For communication, it uses the Dataspace Protocol (HTTPS) and operates with out-of-band transfer protocols. A graphical user interface is available and is protected with user accounts, providing interfaces for users, management, and administration. The sovity CaaS **has not received IDS Certification**. It also supports identity management but specifics on the type are not provided. The connector does not support the IDS Information Model but integrates with a Catalogue/Meta Data Broker. It does not integrate with a Clearing House.

5.2.4.5 Tech2B SCSN Connector

The Tech2B SCSN Connector [56], **provided by Tech2B**, facilitates ***the digital exchange of supply chain-related messages among companies***. It is designed to ***enable data spaces for SMEs through the Tech2B AppStore***, which allows ***industry-specific applications to be integrated with one-click***, enhancing the core features of Tech2B and promoting data spaces for SMEs. The connector is categorized as an off-the-shelf solution available both as a direct service and integrated into data-related products. A **closed source offering**, it is at a maturity level of **TRL 7-8**, indicating that it is in the late stages of development or initial stages of deployment. It is platform agnostic, with deployment options including on-premises and cloud environments. The service levels offered are as a connector service, platform service, and self-service. For access control, it utilizes OAuth, which is an open standard for access delegation.

The connector supports usage control, but the specific policies are not detailed in the section provided. It employs the Dataspace Protocol (HTTPS) for secure communication but does not support the IDS Information Model or any

specific vocabulary. Additionally, it does not integrate with Catalogue/Meta Data Brokers or Clearing Houses. The connector is not IDS certified. Adoption examples include its use in connecting SMEs to various industry-specific applications through the Tech2B platform, significantly simplifying digital data exchange processes within supply chains. Regarding identity management, the Tech2B SCSN Connector supports decentralized identity management, which enhances security and user control over identity data. This connector is particularly geared towards enabling SMEs without digitization knowledge to participate in secure and standardized data exchanges within supply chains, enhancing their connectivity and future-proofing their operations.

5.2.4.6 Telekom DIH Connector

The Telekom DIH Connector [57], **developed by T-Systems International GmbH**, is designed ***for fast and user-friendly data sharing across various data spaces***. It connects to any data space "in under five minutes", is one of the first connectors **on track for IDS certification**, and complies with Gaia-X standards. The connector is cloud agnostic and emphasizes plug-and-play functionality, which simplifies technical setups and promotes trusted data exchange with enhanced data sovereignty. This connector, an off-the-shelf solution offered as a service, is at a maturity level of **TRL 8** and is available under a **closed-source license**. It supports both on-premises and cloud deployments and functions as Connector-as-a-Service. For access control, it utilizes OAuth and API keys, includes comprehensive usage control policies aimed at data consumers, and features a graphical user interface for ease of use in management and administration tasks.

For communication, it employs the Dataspace protocol (HTTPS) with both in-band and out-of-band data transfers. The connector supports centralized (X.509) and decentralized (did:web) identity management systems. Additionally, it integrates with a Catalogue/Meta Data Broker, supports the IDS Information Model (version 4.2.0), and offers dataspace-specific vocabularies. Significantly, it includes integration with the Gaia-X Digital Clearing House, making it a robust solution for sectors such as Catena-X and Gaia-X for Future Mobility. Adoption examples of the Telekom DIH Connector include its use in enabling robust and secure data exchanges in industrial and research applications, particularly in sectors like automotive manufacturing and smart mobility, as part of the broader Gaia-X framework.

5.2.4.7 Tritom Enterprise Connector

The Tritom Enterprise Connector [58], **managed by DataSpace Europe Oy,** enhances ***technical connectivity between data source and target systems within the Tritom service***, facilitating ***the creation of data and service catalogues based on data sovereignty principles***. It is positioned as an off-the-shelf solution provided as a service and is currently used within the licensed Tritom service. The connector boasts a **TRL of 8**, indicating its advanced stage of readiness and deployment. It operates under a **closed-source Tritom Enterprise license** and is platform agnostic, allowing deployment options on-premises or in the cloud. The service level offered is platform as a service.

For security, it supports OAuth and API key access controls but does not include usage control policies. Communication is facilitated through the REST protocol, although specific transfer protocol information is not provided. The Tritom Enterprise Connector does not support the IDS Information Model and is content agnostic without any specific vocabulary support. It also does not integrate with any Catalogue/Meta Data Broker or Clearing House. Notably, **the connector is not IDS certified**. This connector is part of the ongoing development of the Tritom service version 1.0.

5.2.4.8 Trusted Supplier Connector

The Trusted Supplier Connector [59], **developed by German Edge Cloud GmbH & Co. KG**, is ***designed for usability through its Configuration and Monitoring UI, tailored for cloud, edge, and hybrid scenarios***. This off-the-shelf solution, provided as a service, achieves a maturity level of **TRL 8** and is an IDS-ready component, though **it is not yet IDS certified**. The connector operates on a **closed-source proprietary/individual license** and is platform agnostic. Deployment options for the Trusted Supplier Connector include edge, on-premises, and cloud

environments, offering flexibility in its application. The service levels available are connector as a service, platform as a service, and self-service, catering to various user needs.

It employs API keys for access control but does not support usage control policies, focusing on straightforward integration and operation. The communication protocols used are IDS Multipart, HTTP, and Cloud Events, with IDS Header, handling database access managed by the provider. A graphical user interface is available for both users and administration, enhancing the ease of use. However, the connector does not support identity management, the IDS Information Model, or any vocabulary, and it is not integrated with any Catalogue/Meta Data Broker or Clearing House. Finally, the connector has been validated in a series of projects / initiatives (Fraunhofer HHI, Fraunhofer HHI Digitale Signalverarbeitung, ICNAP IPT-HHI).

5.2.4.9 VTT DSIL Connector (DSILC)

The VTT DSIL Connector [60], maintained by **VTT Technical Research Centre of Finland**, significantly elevates the capabilities of the Dataspace connector reference implementation. It introduces ***advanced features like support for the OPC UA communication protocol, robust user and role-based access management, and enhanced security measures against XSS, clickjacking, and DOS attacks***. Additionally, rigorous validation checks ensure the connector's integrity and reliability, with enhanced security for the Postgres database. This connector is an off-the-shelf solution available both as a service and directly usable integrated in data-related products. It is platform agnostic and currently uses a **closed-source license**, with final decisions on the license yet to be made. The VTT DSIL Connector **has undergone IDS Certification** and is used in several research and development projects including TRUSTEE, OSME, and RESONANCE.

The connector supports deployments on-premises and in the cloud, with service levels of connector as a service and platform as a service. It implements OAuth and Basic auth for access control and supports extensive usage control policies covering data consumer needs, security levels, user roles, and more. For communication, it uses IDS Multipart, IDS protocol (IDSCP), and Dataspace protocol (HTTPS) with in-band with determined protocol bindings. Although it does not have a graphical user interface, it supports centralized identity management (X.509) and integrates with the IDS metadata broker reference implementation. The VTT DSIL Connector supports various versions of the IDS Information Model but does not support any specific vocabulary or integration with a Clearing House.

5.2.5 Mid TRL Connector Exception

5.2.5.1 TRUE Connector

The TRUE Connector [61], **maintained by Engineering Ingegneria Informatica SpA**, is categorized as a **generic open-source solution** within the IDSA framework. It is at a maturity level of **TRL 6** and is participating in the IDSA Graduation Scheme at the Sandbox stage. This connector ***is designed to be highly portable and can be deployed across various environments including edge devices, on-premises, in the cloud, and on IoT/CPS/OT devices.*** This connector operates under an open-source AGPL version 3 license, although **it has not yet achieved IDS Certification**. The TRUE Connector facilitates trusted data exchanges within the IDS Ecosystem. This setup ensures secure, standardized data exchanges and linkages within a trusted business ecosystem.

Additionally, the TRUE Connector is included in the Fiware Catalogue, ensuring seamless integration with existing Fiware ecosystems. This integration is facilitated by a dedicated Data APP that enables IDS-based interactions in a plug-and-play manner. It has been adopted in several research projects such as Market4.0, AI Regio, Platoon, Circular TwAIn, Eur3ka, CLARUS, SCREAM, and CiTrace, demonstrating its versatility and utility in diverse applications. In terms of deployment, the TRUE Connector offers multiple service levels: Connector as a service (CaaS), platform as a service (PaaS), and self-service, providing flexibility depending on the users' technical

expertise and needs. For security, it utilizes Basic authentication for access control and supports a comprehensive set of usage control policies to manage data interactions securely.

Communication through the connector is facilitated by IDS Multipart and IDS protocol (IDSCP), ensuring data is transferred securely within determined protocol bindings. Although it lacks a graphical user interface, which might limit some user interactions, it compensates with strong identity management capabilities, supporting centralized identity verification using X.509 standards. However, the TRUE Connector does not support the IDS Information Model nor it provides its own vocabulary, leaving the responsibility of data management and schema definition to the data applications integrated with it. It is integrated with both Catalogue/Meta Data Broker and Clearing House, specifically with the Fraunhofer Clearing House, enhancing its data governance and interoperability capabilities within the IDS ecosystem.

5.3 Connector Selection

Based on the evaluation and analysis of all the aforementioned dataspace connectors, two choices prevailed. The first selection is the Eclipse Dataspace Components (EDC) Connector. The EDC Connector aligns well with the criteria / factors defined, covering most of the required aspects, and is continuously being enhanced. It conforms to key Dataspace Initiatives such as IDSA and GAIA-X, and is actively utilized in projects like EONA-X and Catena-X. The second selection is Engineering's TRUE Connector. Although the TRUE Connector is involved in several EU research and dataspace projects, its comparatively lower TRL makes it a more challenging option relative to the EDC. However, both Connectors can be tested, in order to achieve compliance with at least five (5) CEDS initiatives.

5.4 Next Steps

The next phase of the current task involves a detailed testing of the Eclipse Dataspace Components (EDC) connector within a Minimum Viable Dataspace (MVD) setting, followed by potential modifications to the existing code to enhance functionality and compatibility. Upon finalizing the connector software, the major stage of the integration process with the Common European Data Spaces (CEDS) will be initiated. However, this stage presents certain challenges that need addressing. One challenge is the potential additional testing of the TRUE Connector, which must be a separate task from that of the EDC Connector testing. In addition, what needs to be clarified is which aspects of Data Governance will manage the data exchange between CEDAR's Data Space. These issues are crucial for ensuring seamless interoperability and maintaining data integrity across the CEDAR platform.

6 Cybersecurity

Reliability and robustness of the CEDAR technology will heavily depend on the security of the **(1)** developed and integrated SW artefacts (including ML algorithms, ML models, SW components, APIs, CEDS connectors, and user interfaces), **(2)** generated and (re)used data sources, and **(3)** deployment environments and configurations. To this end, we will conduct a **(manual) continuous cybersecurity risk assessment** of the CEDAR assets, and accordingly propose technical and organisational measures that will drive the design, development, deployment, and use of the CEDAR solutions. Furthermore, to enable their real-time cybersecurity monitoring, we will develop, integrate, and utilise a **network intrusion detection system (nIDS)** following the requirements and specifications defined in WP1. Finally, as the CEDAR technology matures and moves into relevant environments, we will also conduct **penetration tests**.

In doing all this, we will follow several standards and industry best practices for cybersecurity risk management, including:

- **ISO/IEC 27001:2022** "Information Security, Cybersecurity and Privacy Protection — Information security management systems – Requirements". [62]
- **ISO/IEC 27005:2022** "Information Security, Cybersecurity and Privacy Protection — Guidance on Managing Information Security Risks". [63]
- **NIST SP 800-37 Rev.2** "Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy". [64]
- Best practices on risk management from **ENISA**. [65]
- Best practices on threat modelling from **OWASP** [66] and **MITRE ATTACK**. [67]

CEDAR relies on different AI technologies. To ensure **technical robustness of AI** (i.e., resilience to training and inference attacks, accuracy, reliability, reproducibility), we will further consider specific AI-related guidelines for threat modelling, risk assessment, and overall risk management. Some examples include:

- **NIST AI 100-1** "Artificial Intelligence Risk Management Framework (AI RMF 1.0)". [68]
- **NIST AI 100-2 E2023** "Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations". [69]
- **MITRE ATLAS** "Adversarial Threat Landscape for Artificial-Intelligence Systems". [70]

In the remainder of this section, we present our **cybersecurity risk assessment framework** (Section 6.1), and a brief description of our approach to **network intrusion detection** (Section 6.2) and **penetration testing** (Section 6.3) that will be performed at later stages of the project and reported in detail in the subsequent WP3 deliverables.

6.1 Cybersecurity Risk Assessment Framework

We will take an **asset-based risk assessment approach**, in which we perform the following steps:

1. Identify **assets** developed and/or used in the context of CEDAR. This includes:
    a. Datasets (research data generated or reused in CEDAR, system and network data).
    b. AI models.
    c. Data sources and destinations (sensors, databases, etc.).
    d. Data processing components (algorithms, components, libraries, etc.).
    e. Interfaces (APIs, CEDS connectors, user interfaces, etc.).
2. Identify **threats and vulnerabilities** to/within the assets identified in step #1.
3. Identify, evaluate, prioritize **risks** based on the results of step #2.
4. Propose **technical and organisational measures** to manage the risks identified in step #3.

In terms of the scope of our risk assessment, we will consider **3 key assessment dimensions / layers**:

1. **Inter-spection**: we consider the complete CEDAR architecture, and we assess the security of the **information/data flows**, i.e., the interactions among the CEDAR assets. In practice, this means performing risk assessment on the results of WP1, specifically of the task T1.3 providing the technology design.
2. **Intra-spection**: we focus on the **individual assets**, and we assess the security of their structure, dependencies, and functionalities. In practice, this means performing risk assessment on the results of WP2-WP4 that deliver technologies for data preparation, management, and analysis.
3. **En-spection**: we focus on the different **environments** in which the CEDAR assets will be deployed, configured, and used by different users with different roles, rights, and needs, while considering the different **contexts** in which CEDAR will be used. Note that CEDAR will be demonstrated and validated in 3 different pilots through different use cases in which different sets of CEDAR tools, functionalities, and datasets will be used, which will be subject to different threat landscapes.

For each layer, the assessment will be done following several steps, as described below.

STEP #1. ASSETS: identify all key assets developed and/or used in the context of CEDAR. For each asset, identify the following:

- ASSET: ID, name, and short description of the **asset**, as well as the asset **owner** (the partner developing or providing the asset).
- IMPACTS: description and rating of the **impact** in case the asset is, in some way, compromised.

For the impact rating, we use a Low – Medium – High scale.

STEP #2. THREATS: identify relevant threats for all identified assets. For each threat, identify the following:

- ACTORS: type of the potential **attacker** (internal vs. internal actors), their **motivation** (and motivation rate), and their **capability** (and the capability rate) to perform the attack while considering the technology, skills, time, and financial resources required for such action.
- THREATS: name and description of the **threat**, i.e., the way in which the attacker could compromise the system.
  We consider different threat models such as STRIDE [71] and LINDDUN [72].

For the motivation rate and capability rate, we use a Low – Medium – High scale.

Based on the motivation and capability rate, we determine the **threat rate** as shown in the table below.

Threat Rate		Motivation Rate		
		Low	Medium	High
Capability Rate	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High

STEP #3. VULNERABILITIES: identify vulnerabilities in the assets. For each vulnerability, identify the following:

- VULNERABILITIES: name and description of the **vulnerabilities** that could be exploited.
- EXPOSURE: rate of the **ease of the exploitation**, i.e., the level of effort that would be required to exploit the vulnerability, and of the **exposure to threat**, i.e., the level of exposure the vulnerability is subject to.

For the ease of exploitation and exposure to threat, we use a Low – Medium – High scale.

Based on the ease of exploitation and exposure to threat, we evaluate the **vulnerability rate** as shown below.

Vulnerability Rate	Exposure to Threat		
	Low	Medium	High

Ease of Exploitation	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High

STEP #4. LIKELIHOODS: combine the analysis of threats and vulnerabilities to determine how likely it is that a particular threat would make use of a particular tactic or technique to exploit a vulnerability. Based on the previously determined threat rates (in step #2) and vulnerability rates (in step #3), we determine likelihoods as shown in the table below.

Likelihood		Vulnerability Rate		
		Low	Medium	High
Threat Rate	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High

STEP #5. RISKS: combine our analysis of likelihoods (in step #4) with initially determined impacts (in step #1) to define risks as shown in the table below.

Risk Rate		Impact		
		Low	Medium	High
Likelihood	Low	Low	Low	Medium
	Medium	Low	Medium	High
	High	Medium	High	High

STEP #6. COUNTERMEASURES: when risks are clear, we identify existing technical and organisational measures in place to minimise them. Moreover, if possible, available, and needed, we propose additional technical and organisational measures for further enhancing security of the associated assets.

6.2 Real-Time Network Intrusion Detection System

To detect and prevent malicious attacks against CEDAR's communication network infrastructure (communicating devices and communication links), a network Intrusion Detection System (IDS) will be developed by leveraging and extending Sigmo-IDS platform. [73] In particular, the proposed solution will utilize ML techniques to dynamically detect unknown attacks from behavioural network analysis and proactively reconfigure the network to thwart the detected threats. To this aim, a multi-probe decentralized IDS architecture will be considered as seen in Figure 24. In each probe, a Neural-network-based "protocol-aware" intrusion detection operation will be implemented.

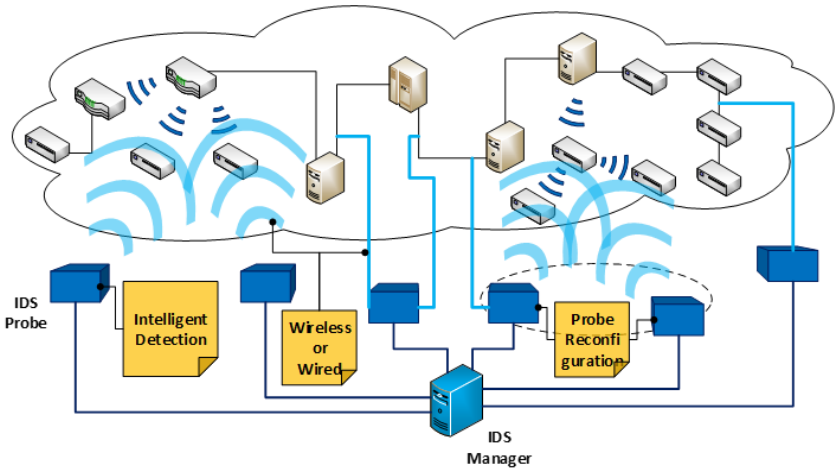

Figure 24. Sigmo-IDS's High-Level Multi-Probe Architecture

Intrusion detection systems are used in today's computer networks to detect and prevent network intrusions. However, because of the ever-increasing sophistication of new/unknown cyber-attacks against communication networks, new IDS systems are being explored; not only to counter such new attacks but also to accommodate the underlying network attack patterns shaped/driven by the increasing usage of ML algorithms including neural networks, which are paving the way to new generations of cognitive/self-adaptive, and potentially collaborative, attacker agents.

IDS systems for communication networks can be classified into two main categories:

(1) Signature-based IDS: it relies on known attack signatures or patterns (typically, a database of pre-configured/pre-defined attack signatures or patterns). This IDS category has several advantages, including simplicity and ease of management. However, it also has limitations, including its reliance on known attack signatures, its inability to process encrypted traffic, its inefficiency to detect new/unknown attacks, and its high associated risk of false negatives.

(2) Anomaly-based IDS: it relies on identifying deviations from previously established baselines of normal behaviour rather than comparing network traffic or system events to a pre-defined set of attack signatures. This is an attractive approach for identifying previously unknown attacks or zero-day attacks as it can adapt to changes in the environment and adjust the baseline of normal behaviour, accordingly, making it more effective in detecting new types of attacks or changes in attack patterns. Nonetheless, current anomaly-based IDS has some limitations including its vulnerability to false positives while resulting in an excessive resource usage (storage, processing power, bandwidth) to continuously monitor and analyse network traffic. This makes it difficult to deploy in large-scale networks.

To address these issues, recent efforts have been made in ML for IDS aimed at increasing the efficiency of anomaly-based IDS in detecting new or unknown threats (i.e., threats with unknown patterns) in real-time. Such ML techniques include neural networks and support vector machines [74]. Finally, with the increasing concerns about data privacy, the research on IDS systems started to explore new techniques to protect sensitive data while still maintaining effective threat detection [75]. Privacy-preserving IDS solutions use techniques such as data anonymization and encryption to ensure that the privacy of sensitive information is not compromised.

In CEDAR, a new IDS solution will be developed by exploring state-of-the-art techniques [76] [77]. Such techniques will be exploited and assessed in the perspective of ensuring real-time detection of new/unknown threats while minimizing both ML-based false positives and network resource usage. Furthermore, the proposed network IDS solution will address the data privacy problem in the data management lifecycle of the ML model to be developed

in the project. To this aim, a privacy-by-design paradigm will be explored, with the objective of achieving the best trade-off between privacy and solution overhead (e.g., computation and memory footprint). A detailed description of the network IDS architecture will be provided in D2.1.

6.3 Penetration Testing

Penetration testing, also known as pen testing, is a crucial process for evaluating the security of platforms, applications, and networks by simulating cyberattacks. This method helps identify vulnerabilities that malicious actors could exploit. Penetration testing involves various techniques and methodologies, primarily categorized into three main approaches: White Box, Grey Box, and Black Box testing.

White Box Testing: also known as clear box or glass box testing, this approach involves a comprehensive assessment of the internal structure, design, and implementation of the platform. The tester has full knowledge of the system, including source code, architecture documentation, and access to internal resources. This approach enables a detailed analysis of code for security vulnerabilities, evaluation of architectural and design flaws, and testing of internal interfaces and data flow. The process includes reviewing and analyzing source code, performing static code analysis to identify common vulnerabilities, and using both manual reviews and automated tools to detect issues. The findings are validated by attempting to exploit identified vulnerabilities in a controlled environment.

Grey Box Testing: this is a hybrid approach where the tester has partial knowledge of the system, which may include access to certain documentation, architecture diagrams, and limited information about the platform's internal workings. This approach allows for a mix of internal and external testing techniques and the identification of vulnerabilities that may not be visible in Black Box testing. Testing is conducted by using provided documentation and partial knowledge to identify key areas of concern, performing targeted testing based on known elements of the system, and employing both automated tools and manual techniques to explore potential vulnerabilities. The findings are validated by simulating real-world attack scenarios that exploit both known and unknown aspects of the platform.

Black Box Testing: this approach involves evaluating the security of a platform without any prior knowledge of its internal workings. The tester simulates an external attack, mimicking the behavior of a real attacker who has no insider information. This approach includes an external vulnerability assessment, network and application penetration testing from an outside perspective, and exploits discovered vulnerabilities to understand their impact. The process starts with reconnaissance, gathering information about the target system through publicly available sources and passive scanning. This is followed by scanning with automated tools to identify open ports, services, and potential vulnerabilities, attempting to exploit identified vulnerabilities to gain unauthorized access or extract sensitive data, and finally documenting the findings, including the vulnerabilities discovered, methods used, and potential impacts on the system.

Scope of the Penetration Testing in CEDAR: initially, we will conduct a risk assessment as described in Section 6.1. This preliminary evaluation will provide a comprehensive overview of the security posture of each CEDAR deployment (one for each of the three pilots). Since penetration testing cannot be performed until all three deployments are in their operational state, the actual testing will be carried out towards the end of the project. At that point, we will select the most mature deployment (also in alignment with the exploitation plan developed in WP6) and perform a detailed penetration test on it. This detailed test will involve a White or Grey Box assessment, depending on the defined IPR, providing deep insights into the internal structure and implementation of the CEDAR platform and/or specific components. Such penetration testing will help us thoroughly evaluate potential security vulnerabilities, ensuring the robustness and reliability of CEDAR.

7 Conclusion

In close collaboration with WP2 activities (identification of the suitable (available and needed) datasets), this document presented the initial work for the collection of data (initially focusing on the SotA analysis).

In particular, it started the design of a DataOps infrastructure based on the CRISP-DM model, and the design of MLOps for data preparation, model deployment and monitoring.

Moreover, this document described an initial analysis of the CEDS and their potential integration in CEDAR to increase the availability of data for socio-economic applications across the EU.

The last section of the document presented how to conduct a (manual) continuous cybersecurity risk assessment of the CEDAR assets following all the security requirements and specifications defined in WP1.

8 List of References

[1] "CRISP-DM," [Online]. Available: https://www.datascience-pm.com/crisp-dm-2/.

[2] J. C. (. R. K. (. T. K. (. T. R. (. C. S. (. a. R. W. (. Pete Chapman (NCR), CRISP-DM 1.0 Step-by-step data mining guide, SPSS.

[3] C. C. Aggarwal, Data Mining: The Textbook, Springer, 2016.

[4] "IDSA Data sovereignty," [Online]. Available: https://internationaldataspaces.org/why/data-sovereignty/.

[5] "International Data Spaces," [Online]. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/v/ids-ram-4/introduction/1_1_goals_of_the_international_data_spaces.

[6] C. B. F. H. F. K. Michael R. Berthold, Guide to Intelligent Data Analysis: How to Intelligently Make Sense of Real Data, Springer, 2010.

[7] "Ydata Profiling," [Online]. Available: https://docs.profiling.ydata.ai/.

[8] R. C. G. a. R. E. Woods, Digital Image Processing, Gatesmark: 3rd ed. Knoxville, 2007.

[9] D. A. F. a. J. Ponce, Computer Vision a Modern Approach, Pearson.

[10] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022.

[11] "Imputation of missing values," [Online]. Available: https://scikit-learn.org/stable/modules/impute.htm.

[12] "Three Approaches to Encoding Time Information as Features for ML Models," [Online]. Available: https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/.

[13] "Feature Selection," [Online]. Available: https://scikit-learn.org/stable/modules/feature_selection.html.

[14] "Smart Data Models," [Online]. Available: https://github.com/smart-data-models.

[15] W. F. a. all, "Graph Machine Learning in the Era of Large Language Models," 2024. [Online]. Available: https://arxiv.org/abs/2404.14928.

[16] N. G. Yaron Haviv, Implementing MLOps in the Enterprise, O'Reilly.

[17] "Apache Airflow," [Online]. Available: https://airflow.apache.org/docs/apache-airflow/stable/.

[18] "Apache Beam," [Online]. Available: https://beam.apache.org/documentation/.

[19] "Apache Flink," [Online]. Available: https://nightlies.apache.org/flink/flink-docs-stable/.

[20] "Apache NiFi," [Online]. Available: https://nifi.apache.org/documentation/v2/.

[21] D. K. N. &. H. S. Kreuzberger, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," 2022. [Online]. Available: https://arxiv.org/abs/2205.02302.

[22] "Docker," [Online]. Available: https://docker.com.

[23] "Kubernetes," [Online]. Available: https://kubernetes.io/.

[24] "Argo Workflows," [Online]. Available: https://argoproj.github.io/workflows/.

[25] "Kubeflow," [Online]. Available: https://kubeflow.org/.

[26] "MLFlow," [Online]. Available: https://mlflow.org/.

[27] "Prometheus," [Online]. Available: https://prometheus.io/.

[28] "Graphana," [Online]. Available: https://grafana.com/.

[29] "Jenkins," [Online]. Available: https://www.jenkins.io/.

[30] "GitHub," [Online]. Available: https://github.com/.

[31] "GitLab," [Online]. Available: https://about.gitlab.com/.

[32] "Argo CD," [Online]. Available: https://argo-cd.readthedocs.io/.

[33] "Data Version Control - DVC," [Online]. Available: https://dvc.org/ .

[34] "Pachyderm," [Online]. Available: https://pachyderm.com/.

[35] "Amazon SageMaker," [Online]. Available: https://docs.aws.amazon.com/sagemaker/.

[36] "Jupyter," [Online]. Available: https://jupyter.org/.

[37] "Weights and Biases," [Online]. Available: https://wandb.ai/.

[38] "TensorFlow," [Online]. Available: https://tensorflow.org/.

[39] "Apache Kafka," [Online]. Available: https://kafka.apache.org/.

[40] "React Flow," [Online]. Available: https://reactflow.dev/.

[41] "React," [Online]. Available: https://react.dev/.

[42] "Seldon Core," [Online]. Available: https://docs.seldon.io/projects/seldon-core.

[43] *IDSA Data Connector Report,* https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Data-Connector-Report-89-No-13-March-2024-5.pdf.

[44] *Huawei Boot-X Connector,* https://www.boot-x.eu/.

[45] *ECI Gatewise Solutions,* https://gatewise.ecisolutions.com/.

[46] *Eclipse Dataspace Components (EDC) Connector,* https://github.com/eclipse-edc/Connector.

[47] *OneNet Connector,* https://www.onenet-project.eu//onenet-connector-included-in-the-idsa-data-connector-report/.

[48] *Sovity EDC Framework,* https://edc.docs.sovity.de/.

[49] *Sovity EDC Extensions,* https://github.com/sovity/edc-extensions.

[50] *TNO Security Gateway,* https://tno-tsg.gitlab.io/.

[51] *Smart Connected Supplier Network,* https://smart-connected-supplier-network.gitbook.io/processmanual/.

[52] *Trusted Connector,* https://github.com/Fraunhofer-AISEC/trusted-connector.

[53] *AI.SOV Connector,* https://ai-sov.eu/.

[54] *GDSO Connector,* https://gdso.org/Home.

[55] *Dat4Zero Project Website (where Kharon Connector was developed),* https://dat4zero.eu/work-packages/.

[56] *Tech2B Connector,* https://app.tech2b.cc/apps/6/SCSN Connector/SCSN.

[57] *Telekom DIH Connector,* https://internationaldataspaces.org/t-systems-and-idsa-achieve-milestone-for-data-spaces-first-certification-of-a-connector-promotes-standardization-and-interoperability/.

[58] *Tritom Enterprise Connector,* https://www.dataspace.fi/en/data-intermediation-service.

[59] *Trusted Supplier Connector,* https://gec.io/solutions/gaia-x-dienste/.

[60] *VTT DSIL Connector,* https://www.idsa-finland.fi/vtt-has-officially-kicked-off-the-certification-process-for-their-ids-connector/.

[61] *TRUE Connector,* https://github.com/Engineering-Research-and-Development/true-connector.

[62] "ISO/IEC 27001:2022," [Online]. Available: https://www.iso.org/standard/27001 .

[63] "ISO/IEC 27005:2022," [Online]. Available: https://www.iso.org/standard/80585.html .

[64] "NIST SP 800-37 Rev.2," [Online]. Available: https://csrc.nist.gov/pubs/sp/800/37/r2/final .

[65] ENISA, "Best practices on risk management," [Online]. Available: https://www.enisa.europa.eu/topics/risk-management .

[66] OWASP, "Threat modelling," [Online]. Available: https://owasp.org/www-project-threat-model/ .

[67] "MITRE ATTACK," [Online]. Available: https://attack.mitre.org/ .

[68] "NIST AI 100-1," [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1 .

[69] "NIST AI 100-2 E2023," [Online]. Available: https://csrc.nist.gov/pubs/ai/100/2/e2023/final .

[70] "MITRE ATLAS," [Online]. Available: https://atlas.mitre.org/ .

[71] "STRIDE threat model," [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html .

[72] "LINDDUN threat model," [Online]. Available: https://linddun.org/threat-types/ .

[73] "CEA Sigmo-IDS platform," [Online]. Available: https://list.cea.fr/en/page/sigmo-ids-a-software-solution-for-secure-communications/ .

[74] M. H. e. al., "Network intrusion detection system: A survey on AI-based techniques," *Experts Systems,* vol. 39, no. 9, 2022.

[75] A. T. e. al., "FEDGAN-IDS: Privacy-preserving IDS using GAN and Federated Learning," *IEEE Comp. Comm.,* vol. 192, no. C, 2022.

[76] K. R. e. al., "ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model," *Nature Sci. Rep.,* vol. 12, 2022.

[77] D. P.-P. e. al., "Unveiling the potential of GNNs for robust Intrusion Detection," *ACM SIGMETRICS PER,* vol. 49, no. 4, 2022.

[78] I. Dm-Crisp, "Dm-Crisp," [Online]. Available: https://www.ibm.com/docs/en/spss-modeler/18.4.0?topic=dm-crisp-help-overview.

[79] J. H. Rüdiger Wirth, "CRISP-DM: Towards a Standard Process Model for Data Mining".

[80] P. C. Clifton, "Introduction to Data Mining," [Online]. Available: http://www.cs.purdue.edu/homes/clifton/cs490d/Process.ppt .