Ref. Ares(2024)3270359 - 04/05/2024

DAI-DSS INFRASTRUCTURE AND SETUP REPORT AT USE CASE SITE

D5.2

Editor Name	Roland Sitar (FLEX)
Submission Date	April 30, 2024
Version	1.0
State	Final
Confidentially Level	PU

Co-funded by the Horizon Europe
Framework Programme of the European Union

EXECUTIVE SUMMARY

The purpose of the document, "D5.2 – DAI-DSS INFRASTRUCTURE AND SETUP REPORT AT USE CASE SITE" is to provide an update on the progression and current state of implementation on use case sites of the Democratic AI-based Decision Support System (DAI-DSS) Architecture as part of the FAIRWork project.

Section 2 of the deliverable describes the overall infrastructure setup process at the use case partner side. It also describes the setup of the isolated computing system for the DAI-DSS tool to avoid risks to the corporate network.

Section 3 then details the technical preparation tasks such as data collection, human expert training, user selection and implantation of the current DAI-DSS prototype as a use case site and integration with the legacy systems.

Section 4 describes the overall testing procedure. This includes a brief description of the user evaluation and the corresponding KPIs for testing the use cases in general.

In summary, this deliverable provides a brief update on the deployment of the first iteration of the FAIRWork DAI DSS. The first iteration has clearly shown that the DAI-DSS is a useful tool for future decision making in Flex and CRF.

PROJECT CONTEXT

Workpackage	WP5: Demonstration of FAIRWork at Use Case Site
Task	D5.2: DAI-DSS INFRASTRUCTURE AND SETUP REPORT AT USE CASE SITE
Dependencies	This report documents the outcome of the WP5 DAI-DSS concepts for infrastructure and installations as well as summarizes the setup, Data integration Legal aspects including data anonymization

Contributors and Reviewers

Contributors	Reviewers
CRF FLEX MORE JOTNE	Herwig Zeiner/ JR Rishyank Chevuri/ Jotne Gustavo Vieira /More

Approved by: Robert Woitsch (BOC), Coordinator

Version History

Version	Date	Authors	Sections Affected
1.0	April 30, 2024	Sitar (FLEX)	All

TABLE OF CONTENT

1 INTRODUCTION

The purpose of the document, "D5.2 – DAI-DSS INFRASTRUCTURE AND SETUP REPORT AT USE CASE SITE" is to provide an update on the progression and current state of implementation on use case sites of the Democratic AI-based Decision Support System (DAI-DSS) Architecture as part of the FAIRWork project.

Section 2 of the deliverable describes the overall infrastructure setup process at the use case partner side. It also describes the setup of the isolated computing system for the DAI-DSS tool to avoid risks to the corporate network.

Section 3 then details the technical preparation tasks such as data collection, human expert training, user selection and implantation of the current DAI-DSS prototype as a use case site and integration with the legacy systems.

Section 4 describes the overall testing procedure. This includes a brief description of the user evaluation and the corresponding KPIs for testing the use cases in general.

In summary, this deliverable provides a brief update on the deployment of the first iteration of the FAIRWorks DAI DSS. The first iteration has clearly shown that the DAI-DSS is a useful tool for future decision making in Flex and CRF.

2 INFRASTRUCTURE SETUP PROCESS

2.1 Technical preparation tasks

In this section, we describe the technical preparation tasks involved in setting up DAI-DSS infrastructure.

Isolated environment description

As an initial approach, Flex heads towards implementation of an isolated project server for use-case data sharing. The reason for this approach is to decouple data sharing with the use-case partners and risk mitigation of interferences on the running production IT following with this. Also, Flex Global IT policies regarding data security and integrity, as well Flex's responsibility to its customers and employees do not allow any potential hazard to data integrity.

Once the systems planned and provided by the use-case partners will show a robustness in operation as well as stability in being operated by shopfloor employees, it is intended to modify the interface to allow partial access to live data. At the moment of report creation, no outlook can be provided if this action can be done within the project duration of FAIRWork or will be followed as extended scope at a later point in time.

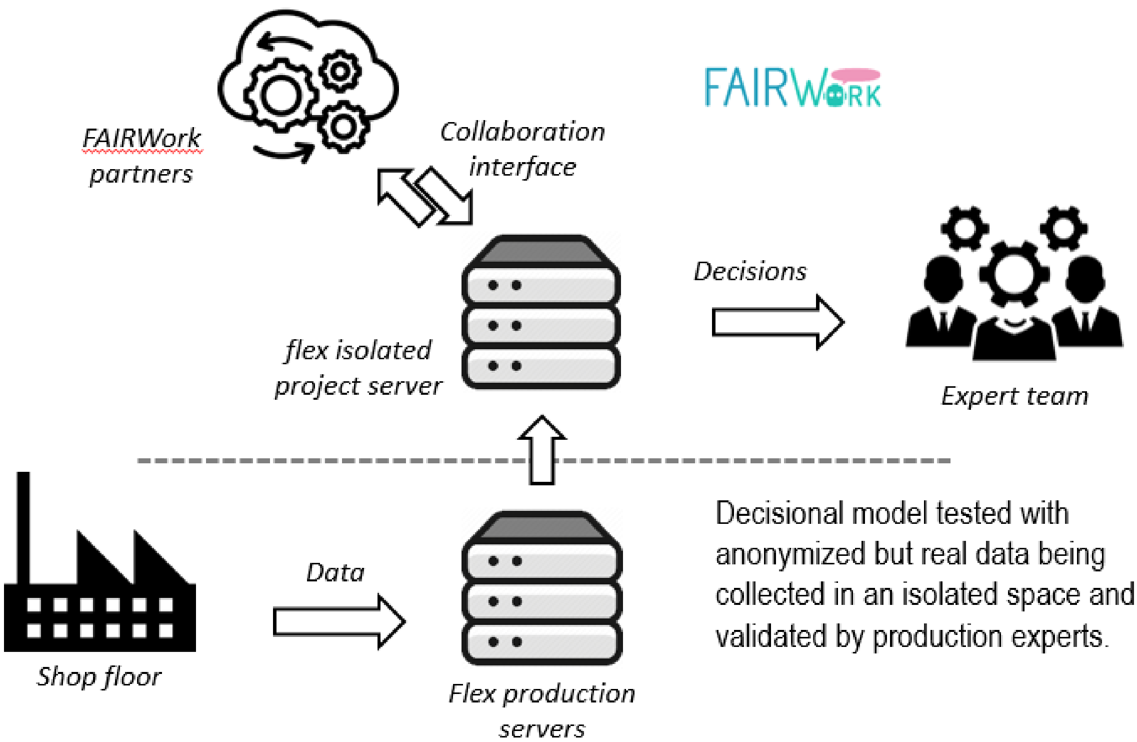


Figure 1: Flex data handling structure

A schematic of the data handling anticipated by the consortium for the Flex use-cases is shown in Figure 1. As already described in the previous paragraph, the centre piece for data transfer will be the "isolated project server" where a copy of live data can be provided to the FAIRWork partners. Initially, the data will be manually uploaded to the services provided by Jotne to speed up the development process of AI algorithms developed by the use-case partners.

The same architecture is also planned for the CRF case study. The two industrial partners have very similar IT security policies, furthermore it is considered preferable to create a platform that can adapt to both industrial environments.

The manually operated interface will allow convenient data handling in the ramp up of the development activities as it offers a graphical user interface to unexperienced users. Independent from use-case scenario, the data upload can be uploaded on a unified file handling tree.

Use-case #1 covering the decision support for the selection of different robotic application scenarios is not focused within this report as it holds challenges that were not solved yet. At the moment, these challenges can be named by the type of input data, which is mainly a pdf document summarizing boundary conditions, and the lack of scenario variations. Summarized, use-case #1 is challenging the implementation by a very low quantity of available data as well as the lack of data that is foreseen to be generated in the near future. The boundary conditions for the individual scenarios are summarized within the pdf documents which only arise if valuable situations on the shopfloor or during development are detected.

Use-case #2 focuses on maintenance and described a breakdown causing standstill times, which should be reduced by support of the DAI-DSS. Due to the project evolution, this use-case was adapted and split into two parts that are applicable to current scenarios on the Timisoara shopfloor.

Part one of use-case #2 focuses on decision support of the operator before contacting a technician of the maintenance staff. The support will be provided by a knowledge base that is currently only used by the technicians and being too complex for the direct use of a new line operator. With the help of the guidance of the DAI-DSS system, the operator’s experience shall be increased more quickly due to the better usability and especially by an implementation of an error search controlled via the operator’s inputs and fault descriptions. If the operators are able to get information from this system, the target is to reduce the efforts of the technicians and maintenance team due to machine breakdowns, as the operators will be supported by various solutions from the new DAI-DSS system.

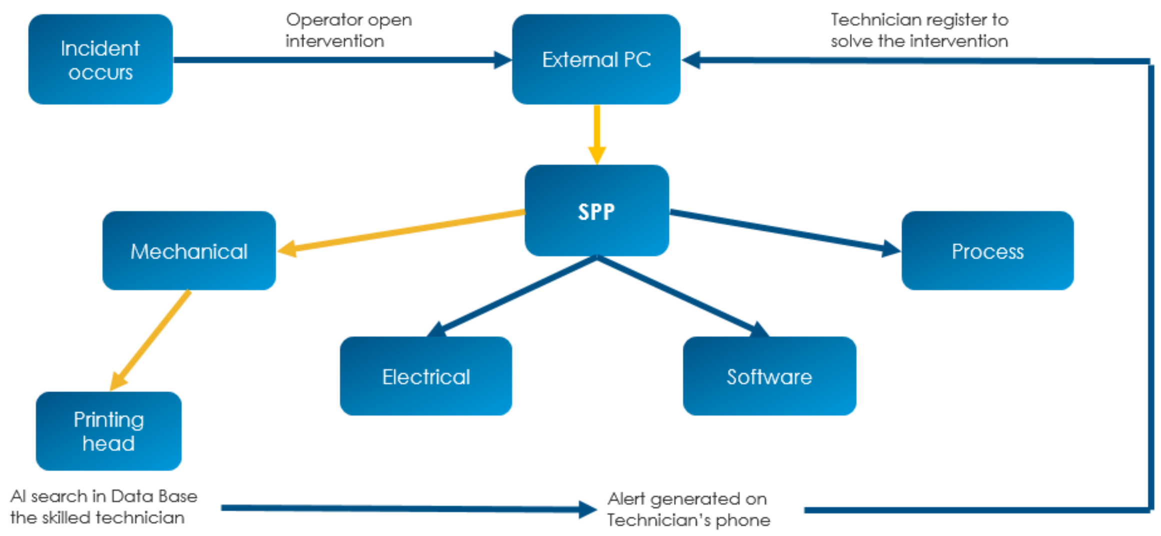


Figure 2: Current workflow for machine breakdown indication

In Figure 2, the current workflow for issuing of a machine breakdown ticket is indicated. Once an incident occurs, the operator must go to an external computer to create a ticket. In this ticket, the operator must describe which specific equipment is affected, select a type of fault (mechanical, electrical, software, process) and then give an indication about potential faults. This information will then be forwarded to the technician or maintenance engineer to start solving this issue, whenever being available.

In the future process, the operator must also go to an external computer, but then only type in the fault description. The DAI-DSS system shall be capable to create links between different keywords being used from the operator (e.g. machine name, specific terms like printing head, component picker, Automatic Optical Inspection, …) and the knowledge base to provide potential solutions that were added to the knowledge base created through past incidents. After checking multiple potential solutions, the operator will still be able to create a ticket to call support from a technician or maintenance engineer. If the issue could be solved by the operator, methods to update the knowledge base with the new solution shall be implemented to let the system learn new wording and solutions.

The effectiveness of this system will be measured through lead time measuring: Incident recorded → incident solved completely and if support by a technician was needed.

Part two of use-case #2 focuses on interpretation of live data from various machines to predict maintenance and therefore reduce or increase the time between maintenance or exchange of components. In first instance, the heating drives of Surface Mount Technology ovens will be equipped with sensors acquiring vibration data. The vibration data will give information about the friction within the heaters and allow an interpretation when maintenance (e.g. lubrication, bearing exchange, …) needs to be applied. To increase the learning curve of the DAI-DSS, it is planned to equip a new and a used oven with sensors to be able to compare the performance. This shall help the decrease to overall project time for this use-case as variations in equipment aging can be analysed directly.

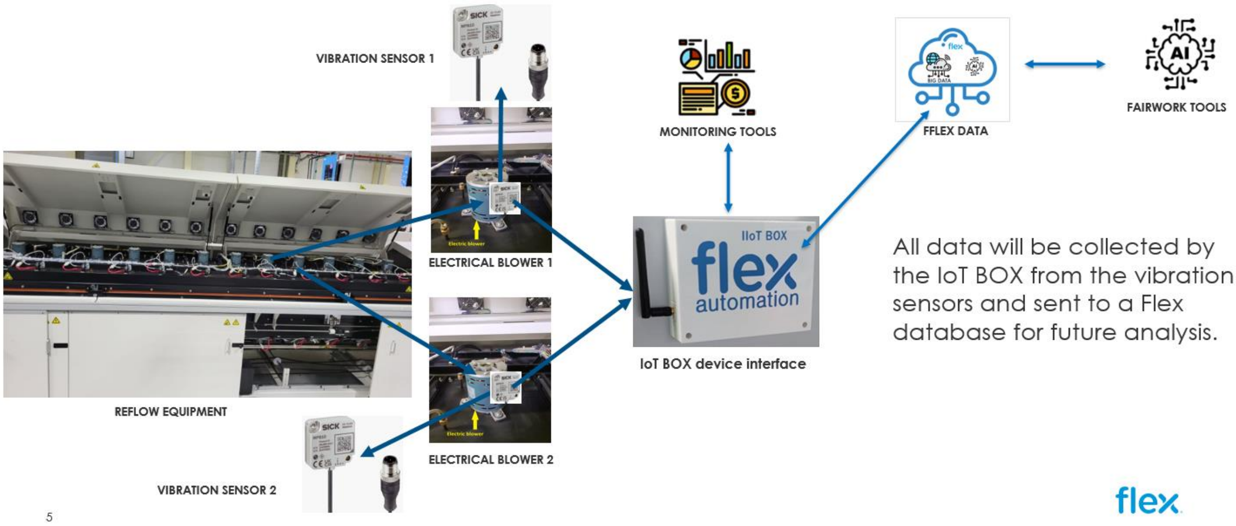


Figure 3: Live data monitoring on SMT equipment

In Figure 3, the aimed setup is indicated by pictures from a test setup. In this setup, Flex will handle the acquired data internally and share it with the project AI database for later usage. In the intial approach, local files will be created which can then be uploaded manually to the AI database for interpretation. Until the end of the project, a handover via a flex isolated project server is planned to share data with the AI platfrom, however a direct access to the data for the FAIRWork project members is not planned as this will be conflicting the IT security policies from flex. Once the system has been implemented and tested successfully for the first equipment, it is also planned to install sensors on other machines of the flex shopfloor to increase the usage of predicitve maintenance. For selection of potential machines it will be important that these are also used on other flex sites to benefit from the initial FAIRWork investigtations and have a quicker implementation as well as commonalities.

The effectivness of this use-case part will be measured by the interventions of the AI driven recommendation compared to the manufacturer's maintenance recommendation. If maintenance intervalls can be increased, this results in direct savings of material since components get a higher life time. If the official maintenance intervall is decreased, a potential break-down can be avoided which leads to savings due to prevented line stand-stills.

Use-case #3 focuses on worker allocaiton and does not require specific hardware to be purchased. In the worker allocation use-case, the target is to get decision support, if an employee is not available as planned on short notice. In this case, the shift manager will be able to request proposals by the DAI-DSS browsing through data like availability, training, rest times, etc. The challenge in this use-case will be the preparation of an interface that will not forward sensitive data to the AI tool, as this would raise additional IT security and data protection tasks. If the data is kept within flex and only shared in an encrypted or masked format, it can be handled by the DAI-DSS without additional measures. Therefore, main focus will be set on flex internal gathering data in one central location, before sharing it with the DAI-DSS and later decoding the response to provide the shift leader proposals, which employee can support.

2.2 Issues and challenge addressed during setup

2.2.1 Cloud offering provided by technical partners

The DAI-DSS Architecture consist of several interconnected components. These components, each a microservice in its own right, engage in continuous communication over the network through the utilization of secure HTTP TCP/SSL protocols, specifically employing REST APIs for their interactions. These communication protocols ensures the integrity and confidentiality of data as it traverses the network, a critical aspect for systems demanding high security standards. The adoption of a microservices-based architecture is a strategy to get the benefits of scalability, flexibility, and resilience. Each microservice is designed to perform a specific function within the DAI-DSS framework, allowing for isolated development, testing, and deployment. This modularity facilitates easier updates, rapid scaling of components based on demand, and enhanced fault isolation, where the failure of a single component does not necessarily compromise the entire system. By leveraging cloud-based solutions, the DAI-DSS architecture can take advantage of cloud-based resources and services. These resources range from scalable computing power and storage solutions to advanced security features and management tools, all provided as part of the cloud infrastructure. This ensures that the system can adapt to varying loads with ease.

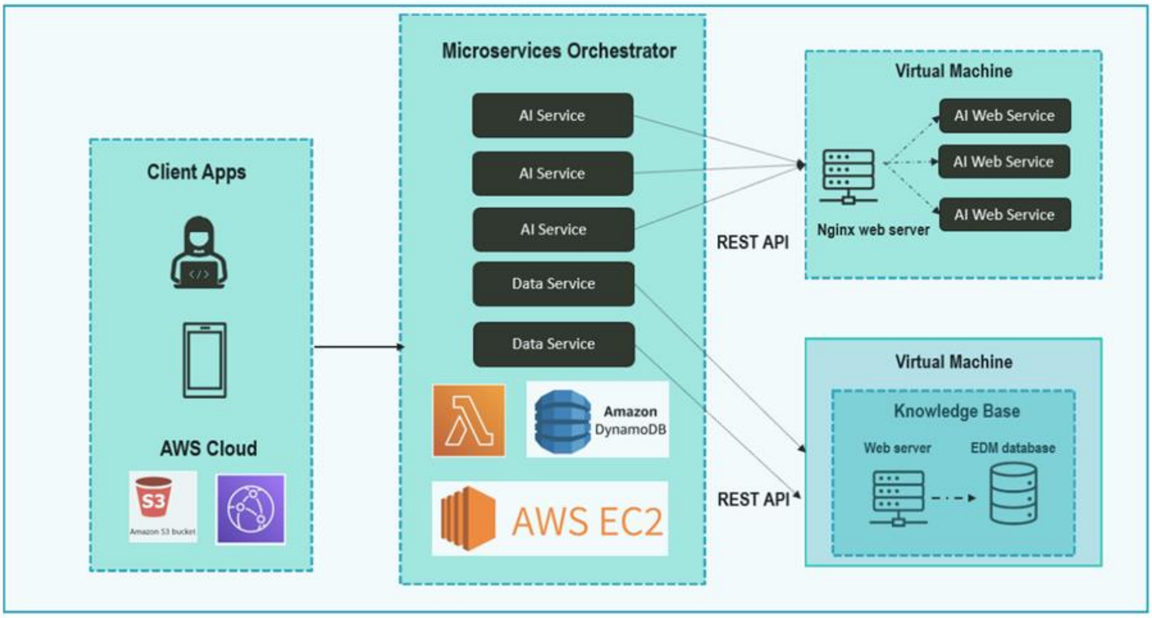


Figure 4: Services overview

In future, if required the installation will be made available at use case sites.

2.2.2 Legacy Data Integration - FLEX

Regarding a dedicated VPN connection for data sharing and access, based on Flex Cybersecurity Team, according to **CSOM Remote Access Procedure – CIT-CIT-3-082-00 Rev:A**, the option to install VPN on flex devices are blocked. Alternative proposals for data exchange are:

- File-Sharing via FTP (flex isolated project server)
  - Flex will place agreed and masked data onto the FTP-Server
  - FAIRWork Partners / DAI-DSS Tool can access Flex-Uploaded data on server
- Access files via API (flex IT-Team provides necessary API to access data)

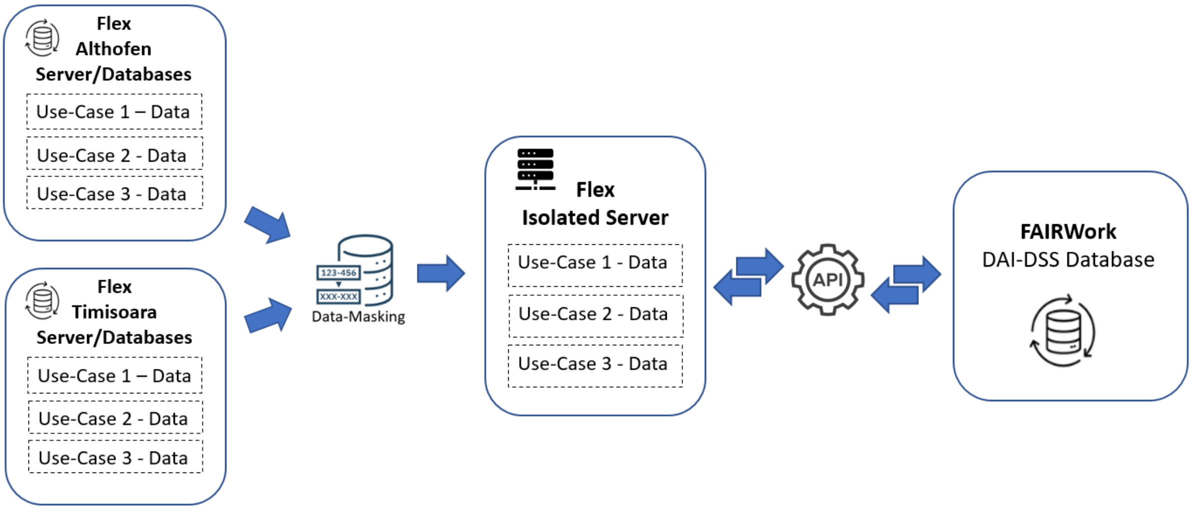


Figure 5: Flex data masking and communication for IT security

Integration of an experimental system leads to additional challenges when being introduced to corporation, especially when focusing on the organization. As global operating company each Flex site has also to follow Flex-Global IT policy and rules.

Before sharing data and setting up the necessary infrastructure the following activities need to be performed:

- Initiate FISDPA approval (Flex Information Security & Data Protection Assessment)
  - Involved FAIRWork Partners will be invited to participate to execute the assessment
- Data Privacy release
  - Based on the required data to share data privacy requirements need to be checked.
  - Data privacy rules will be defined and documented for the project (use-case).
  - Release from global data privacy officer need to be obtained
    - As important input for these activities, all data required to be shared need to be specified.
    - Based on the data-privacy demands "data-masking" requirements can be set (e.g. if masking/coding of personnel ID-Number is required).
- Data access / Data records

-
  - Different databases at use-case sites (Timisoara & Althofen) lead to different standards or tools in operation. For similar records flex needs to find similarities and define common properties or allow future sites attending this project to have an interface to connect to.
  - At the point of report creation not all records can be directly accessed via databases (e.g. individual/manual records still exist within the teams)
    - As different data-sources are used within Flex, a specific connector/linking application will be necessary to allow in future a cross-company knowledge exchange (e.g. Use-case “Machine Maintenance after breakdown”)

The following software challenges were identified, up to the point the report was issued:

- Automated masking / de-masking Software
  - To operate with real-time data an automated masking software will be required before data is uploaded to the Flex Isolated Server / DAI-DSS database.
  - In case personnel data will be masked, a decryption solution to identify suggested operator need to be in place.

All previously highlighted challenges can also be translated to the defined use-cases and are summarized within the next bullet points.

Use-Case #1: Due to organizational re-structuring it is not possible to fully support this use-case from flex.

- no use-case available to gather data
- lack of historical data
- no motion profiles available
- no real-time data available to feedback into DAI-DSS to train AI-Models

based on this limitation FAIRWork partners can only use provided data as uploaded to Nextcloud “*Flex_data_requirements(2).docx*” already. These data can be used from scientific team as playground to simulate fictive scenarios.

Use-Case #2:

- As Flex is a global operating company with multicultural employees, multiple languages for the user-interface and data collection need to be considered.
- Failure description / failure codes might need to be different in Timisoara or Althofen.
  - To share or use the knowledge across the company, translation / linking between the databases will be useful.

2.2.3 Legacy Data Integration - CRF

This section explains how data will be shared towards the “FAIRWork DAI-DSS Database”.

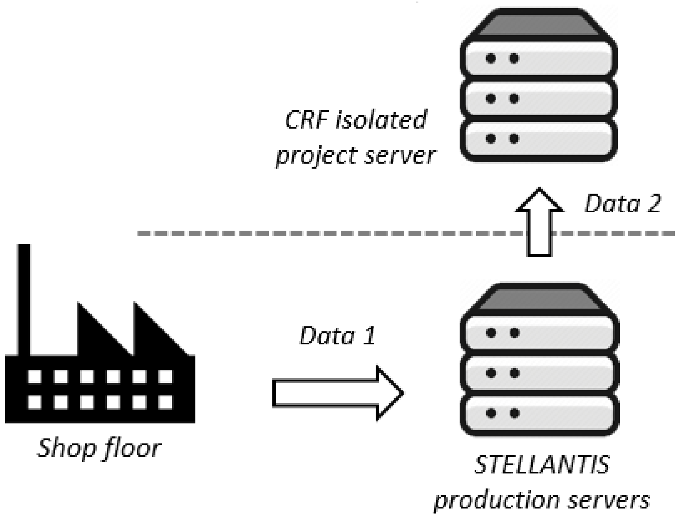


Figure 6: CRF anticipated high level IT structure

As anticipated, the IT security policies do not allow outsiders to access data on the “Stellantis production servers”. With reference to the image above, everything below the dotted line cannot be accessible by external systems.

Access is guaranteed only within the Company network (or through the Company VPN). Access to the network is linked to user’s credentials. Furthermore, access to data is linked to the user's authorization level, this is because some data are considered confidential or strategic.

The transition from "Data 1" to "Data 2" is therefore critical. It will be the responsibility of CRF researchers to populate the "CRF isolated server" with significant data for the project, coming from the "Stellantis production servers".

This data will then be appropriately filtered so as not to create conflict with company policies, for example:

- ***Data referring to people:*** the data used concern the presence, the suitability, and the preference of workers during the work activity. The data must be anonymized, so that it is not possible to trace the identity of the workers.
- ***Data referring production volumes:*** these data are considered non-public and will be modified in a manner consistent with the purposes of the project.
- ***Data referring production quality:*** these data are considered non-public and will be modified in a manner consistent with the purposes of the project.

Regarding the "CRF isolated project server", it is connected to an open network, dedicated to the project and accessible from external systems without particular limitations. The OpenVPN to access to the “FAIRWork DAI-DSS Database” was tested with the Stellantis Guess connection without problems.

Once the data are uploaded to the "CRF isolated project server", synchronization with the "FAIRWork DAI-DSS Database" must be guaranteed.

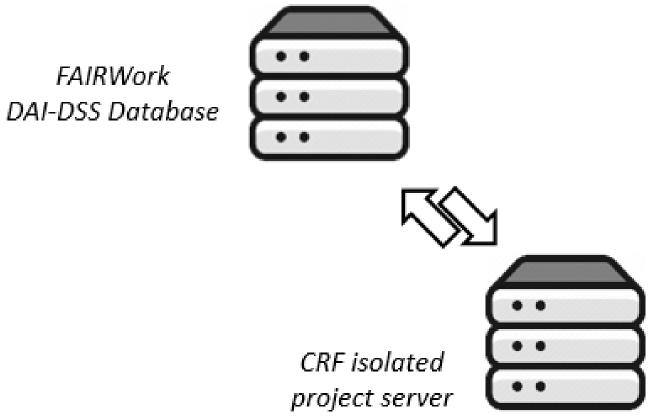


Figure 7: CRF local server to FAIRwork database information exchange

For this purpose, an API that provides the synchronization service will be implemented. It will be resident on the CRF side and will compare the contents of the databases on both servers. If there are new files, they will be uploaded to the "FAIRWork DAI-DSS Database".

The image below schematizes the data flow discussed so far:

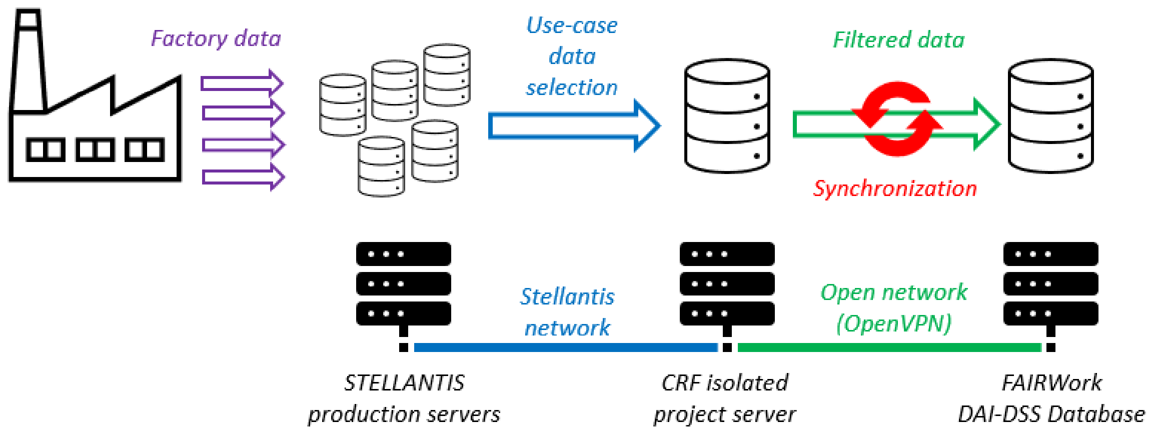


Figure 8: CRF IT structure schematic

The data follows a 3-stage path:

1. The factory stored a multitude of data on the "Stellantis production servers".
2. A selection of these data, useful for the purposes of the project, will be collected and stored in a "CRF isolated server", via the Stellantis network.
3. This data will then be filtered to be compliant with company policies and synchronized with the "FAIRWork DAI-DSS Database", via an open network (using the OpenVPN)

2.3 Use-case organizations

This subsection provides an Overview overview of use case organizations, organizational culture, and skill profiles of involved teams. It describes in short following points.

- Tools / processes
- Assign people to handle tools (e.g., skill matrix)

The proposed tool will be the Breakdown data base. Based on statistics, we can set the base line, for instance, one Engineering Technician takes in average 3months to learn the processes and the equipment until he/she becomes independent. With all the knowledge collected into above mentioned tool, the period is expected to be shorten. Data can be provided only after running the tool. As a KPI we will look at the ratio between Cycle Time & Lead Time.

Example: Training takes 5 days (Ct) / Time between Beginner status & Intermediate status is 90 days. (Lt) actual ratio is 5days/90days= 0.05. If we aim LT=60days new ratio of 0.08

Production staff works in 4 shifts (A; B; C; D) and they are ranked on 3 levels (Beginner, Intermediate, Advanced) based on their experience started with the employment date.

Nr crt	Name	Project_Name	Team / Shift	Department	Job_Name	Program_Type	Employment_Date	Qualification Lvl
1	Operator 1	Varroc Lighting Systems	B	Operations	SMT Operator	12/24/48 - B	30.06.2021	advanced
2	Operator 2	Varroc Lighting Systems	C	Operations	SMT Operator	12/24/48 - C	01.09.2020	intermediate
3	Operator 3	Varroc Lighting Systems	A	Operations	SMT Operator	12/24/48 - A	05.12.2018	advanced
4	Operator 4	Varroc Lighting Systems	D	Operations	SMT Operator	12/24/48 - A	05.10.2023	beginner

Shift pattern is the way of working.

wk48	wk49							wk50						
12/1	12/2	12/3	12/4	12/5	12/6	12/7	12/8	12/9	12/10	12/11	12/12	12/13	12/14	12/15
Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
	A	C	D	B	A	C	D	B	A	C	D	B	A	C
	B	A	C	D	B	A	C	D	B	A	C	D	B	A
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
0	0	0	1	1	0	0	1	1	0	0	1	1	0	0

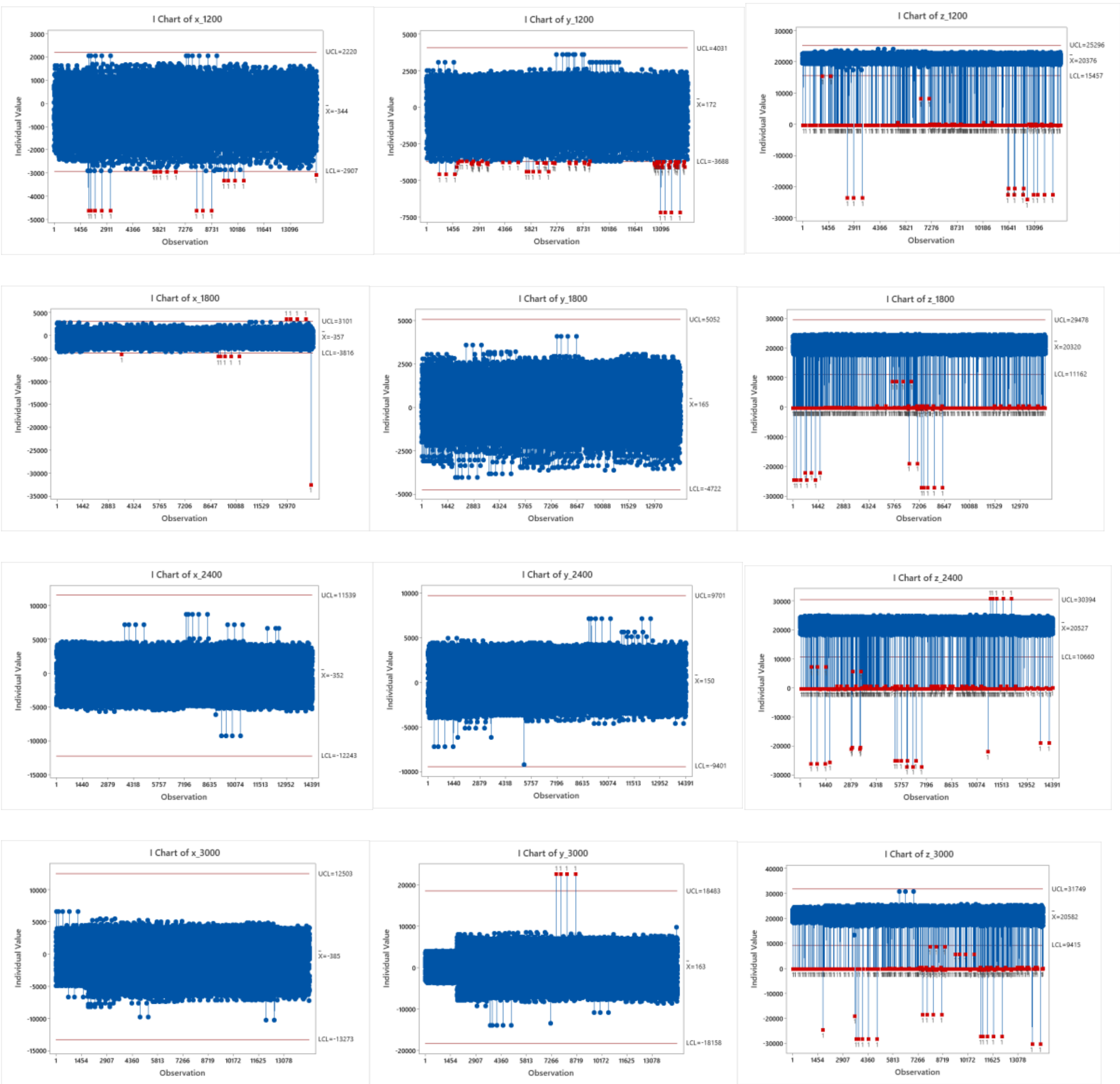
wk51							wk52							wk01	
12/16	12/17	12/18	12/19	12/20	12/21	12/22	12/23	12/24	12/25	12/26	12/27	12/28	12/29	12/30	12/31
Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
D	B	A	C	D	B	A	C	D			C	D	B	A	C
C	D	B	A	C	D	B	A				A	C	D	B	A
0	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1
0	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0
1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1
1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0

3 TECHNICAL PREPARATION TASKS

3.1 Physical data acquisition in use-cases for Flex

The installation of new sensors as needed in the case of the FLEX use cases. For example,

This setup of reflow oven has multiple fans which are working on different parameters, depending on temperatures ranged that need to be controlled during the live production. Bellow pictures with the analyse of “Acceleration” parameter (mechanical offset of the fan’s motor on all axis) on different revolutions per minute (rpm).



All the measurements were done on Lab condition, where we have induced artificially potential issues. The 1st major conclusion it is that, as we increase the rpm, we see less offset on all the axis. Next step is to purchase enough and do trials on pilot Reflow Oven to see if the preliminary conclusions are repeatable.

3.2 Human expert training

Another important point is the training of human expert before carry out the tests with the initial prototype.

1) On boarding (training about safety, fire protection, organization and some specific shallow info about specific job)
2) Intermediate (human becomes aware of what is happening around, at the end of this stage he/she becomes independent, most of the material is process specific)
3) Advanced (human is capable to choose his/her priority, strong knowledge about the processes and equipment assigned)
4) Soft skills (manager decide at which level the staff starts this type of training. Aim is to enhance the soft skill of the staff)
   Most of the trainings are supplied by internal SMEs (Subject Matter Expert) and the rest are provided by 3rd parties (usually equipment suppliers).

There is a bidirectional training request in our system.

- The staff can look for a training in the portfolio and request to his/her manager a specific training.
- The manager can assign individual or team training from the portfolio in order to increase the level of knowledge

3.3 Implementation overview

In this section, we describe in short, the first deployment of the DAI-DSS system

The initial DAI-DSS prototype, detailed in "D4.2 – Initial DAI-DSS Prototype," focuses on integrating fundamental components to provide a structure for the DAI-DSS architecture. This implementation offers a comprehensive view of how various DAI-DSS components, including the User Interface (UI), Orchestrator, Configurator, and Knowledge Base, interconnect and support the decision-making processes within the framework.

User Interface (UI): The UI plays a crucial role in facilitating efficient decision-making by providing a clear visualization of the decision-making process. Using the Workload Balance Use Case Scenario as an example, the UI demonstrates its ability to display critical components directly linked to decision-making, enhancing the overall functionality and value of the system. The implementation and deployment of UI involves setting up the OLIVE framework which plays a critical role in the configuration of the DAI-DSS environment. It includes deploying the Workflow Engine for managing workflows and deploying user interfaces that are crucial for interacting with the system. UI components are deployed in an AWS Bucket and distributed through AWS CloudFront CDN.

DAI-DSS Orchestrator: The Orchestrator is central to the architecture, coordinating the exchange of information and decisions between different parts of the system. It operates through two mechanisms: a Workflow-based Orchestrator, which executes tasks leading to AI service recommendations, and another approach that utilizes agents for decentralized decision-making in a multi-Agent perspective. The process begins with defining workflows that are necessary for tackling different use case scenarios. This involves creating workflows for retrieving required data, such as worker and order information from the knowledge base. The workflow definitions are specified within the workflow engine's configuration environment, where they can be edited or newly added. Each workflow definition includes the tasks to be performed, input parameters needed, and how these tasks interact or depend on each other. Once a workflow has been defined and tested, it can be stored and used across other components of the DAI-DSS system, such as user interfaces or other services and workflows. This integration is facilitated through APIs provided by the OLIVE microservice framework, enabling different interactions with the workflow engine, such as searching, storing, or triggering workflows. The OLIVE framework and the workflow engine are deployed on AWS, utilizing serverless AWS Lambda functions and AWS EC2 instances, relying on AWS DynamoDB for file storage. Workflows are stored in a service that is specific to each sandbox environment, such as the one used for the FAIRWork project, ensuring that workflows are managed and executed within their respective environments.

DAI-DSS Configurator: This tool aids in the efficient configuration and integration of the decision support system. It comprises the Configuration Framework and the Configuration Integration Framework, assisting in the creation of decision models and generating system configurations from these models. The prototype includes a configuration environment that allows for microservices and workflow configuration, as well as a user-friendly interface for combining different UI components.

DAI-DSS Knowledge Base: Acting as a central data repository, the Knowledge Base stores essential data and decision models, supporting various components of the DAI-DSS for configuration and decision-making purposes. It integrates with the Configurator and employs REST API for data retrieval, playing a vital role in the system's data flow. The public version of the Knowledge Base is deployed on the dedicated windows server.

3.3.1 DAI-DSS AI Enrichment:

The Implemented AI services are Hosted on dedicated servers provided by JOANNEUM RESEARCH with specific ports exposed for SSH connections and REST-API endpoints. Utilizes the Anaconda environment manager to manage Python environments and prevent library conflicts. Core functionality of AI services, including models or algorithms, is encapsulated into a REST API executed continuously on the server. Uses the Flask library for establishing REST API endpoints, with additional production deployment requiring a Web Server Gateway Interface (WSGI) implemented using the "Werkzeug" Python library. To simplify firewall configuration on the prototype server, an Apache2 server is installed, with standard ports open for HTTP and HTTPS, forwarding calls internally to the correct server and port.

3.3.2 Deployment of implemented AI services:

3.3.2.1 Rule-based Resource Allocation and Decision-Tree Service deployment

The rule-based service consists of two applications, which are used together. An OLIVE instance and a server for the worker allocation. OLIVE is used to start configured decision services (like the Line Assignment Assessment service) and to wrap the resource allocation service and allow configuring them and making them usable. For example, to pre-configure the endpoint for the line assignment assessment. Both applications are provided as a docker image, which is stored in the GitLab repositories of the corresponding projects.

The decision tree service was deployed as docker image. The docker image is available over the corresponding GitLab project.

3.3.2.2 MAS-based Service

The MAS-based service was implemented in Java, and to fulfil the devised modular approach it was containerised in a docker image. Furthermore, a RESTful API was also developed in Python using the Flask framework, secured with JWT for endpoint protection, to make the output of the MAS service available. This API uses Waitress as a WSGI for production. All elements are behind an Apache proxy server handling external communications. Thus, deployment includes the definition of a set of Docker services, based in docker images, for the MAS-based service, the secure API and the database for credentials information.

The deployment of the DAI-DSS prototype involves various methodologies to integrate the system within the broader architecture. This includes the deployment of the OLIVE Framework, Workflow Engine, User Interfaces, AI services, and a MAS-based service deployment. These components are hosted and made accessible through dedicated servers, leveraging serverless AWS Lambda functions, AWS EC2 instances, and Docker containerization for secure and efficient deployment. The deployment process ensures the system's scalability, security, and compatibility across different software environments, laying the foundation for future enhancement. The final UI is a web application accessible through a URL which provides functionality to invoke the AI services for CRF workload balance scenario.

3.4 DAI-DSS prototype installation at use-case site

Installation of DAI-DSS prototype at use case site

The installation of the DAI-DSS prototype at a use case site involved several key steps to ensure its effective deployment and integration into the existing infrastructure. Figure 9 show the different components required to create a decision support system related to the use case scenarios.

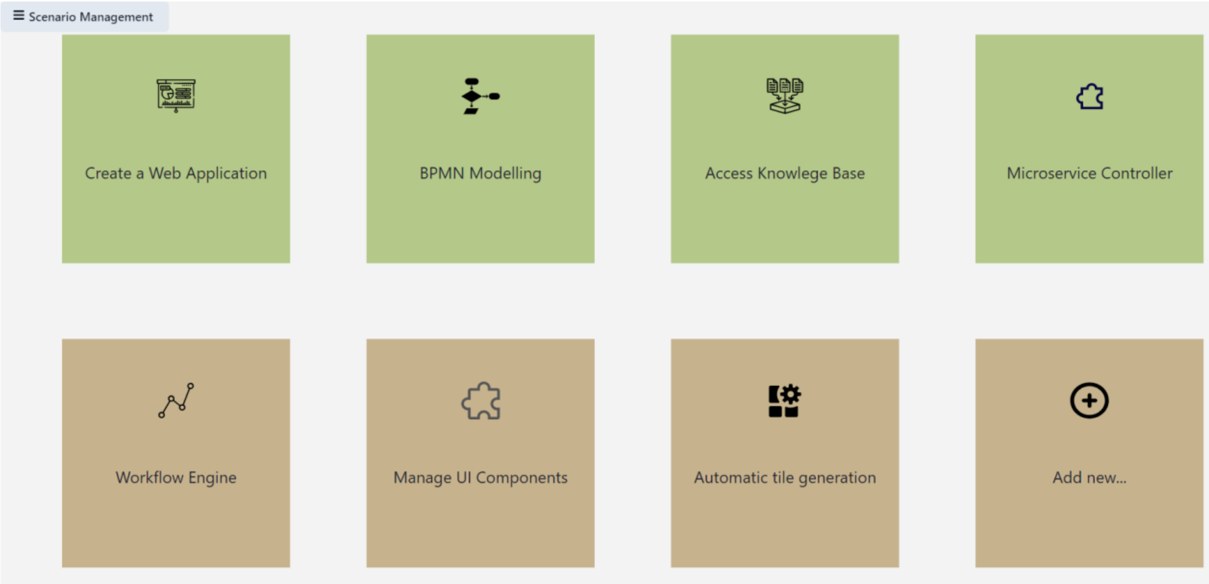


Figure 9 DAI-DSS Configuration environment.

Configurator Implementation: The Configurator tools was instrumental in setting up decision models and integrating them into the system. It provided a framework to assist in configuring decision strategies and generating system configurations based on these models. This included the ability to configure microservices and workflows, enhancing decision support capabilities.Figure 10 show the configuration of microservices coupled with the Knowledge Base REST API methods.

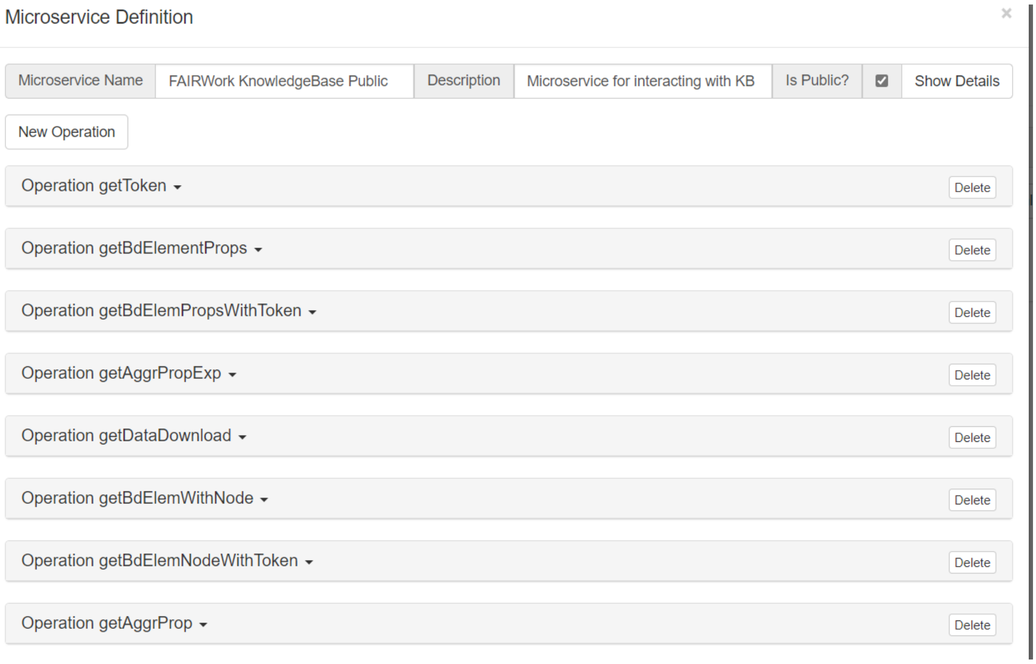


Figure 10 Micro service configurator.

Orchestrator Role: The DAI-DSS Orchestrator played a crucial role in coordinating information exchange and decision-making processes across various components of the system. It ensured that individual services interacted smoothly to retrieve relevant data from the knowledge database. This is managed with the help of creating required workflows that support the CRF workload balance scenario.

Knowledge Base Integration: A central Knowledge Base was established as a repository for storing and retrieving data necessary for decision-making. This included user properties, sensor data, and processed data. The Knowledge Base was integrated with the Configurator and accessible using REST API, crucial for the deployment of AI services and supporting data-driven decision-making.

UI Integration: The User Interface (UI) of the DAI-DSS was integrated to visualize essential decision-making components in the Industrial Partner CRF's Workload Balance Use Case Scenario. This included displaying key indicators and facilitating interaction between decision-makers and the system.Figure 11 show the UI components that are created for to support the required use case scenarios workload balance.

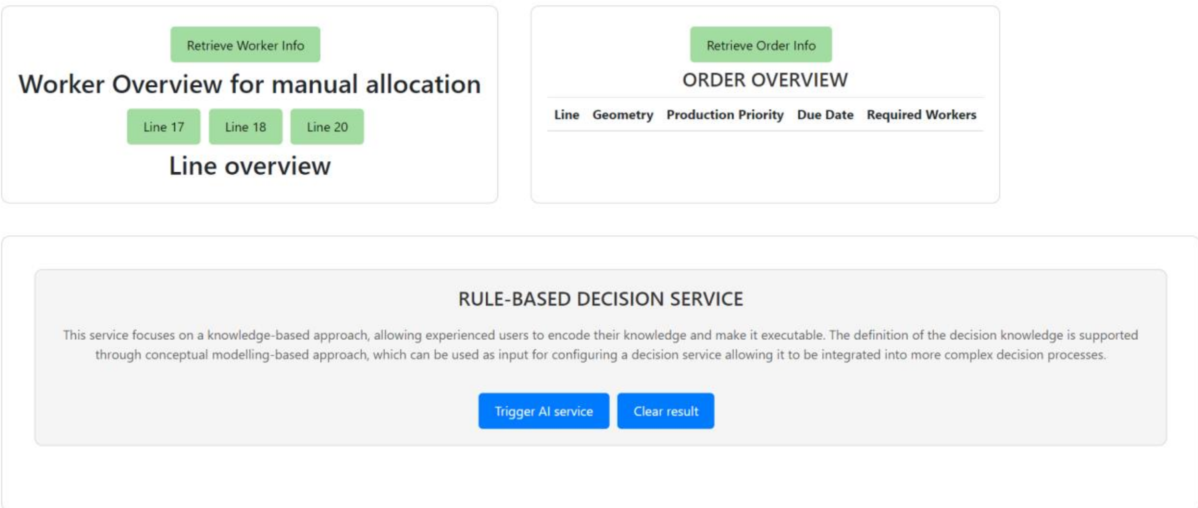


Figure 11 DAI-DSS UI

These steps collectively contributed to the successful installation and integration of the DAI-DSS prototype at the use case site, facilitating efficient and informed decision-making in industrial environments.

3.5 Connectors to legacy systems

The Knowledge Base, hosted at https://demo.jotne.com/EDMtruePLM/#/home , is designed to integrate with legacy systems within the FAIRWork project framework, facilitating efficient data exchange and management. Users from CRF, FLEX gain access through web application using this URL, where role-based permissions guide them to project-specific data management areas. This setup ensures that end users can contribute and manage data effectively, adhering to the structured breakdown element configurations established within the Knowledge Base. To enhance interoperability with legacy systems, the Knowledge Base supports REST API-driven data uploads. This functionality is critical for automating the transfer of operational data, such as operator presence, directly from legacy systems into the Knowledge Base. By utilizing REST APIs, these systems can programmatically send data to the Knowledge Base, where it is processed and stored in a standardized JSON format. This method offers a streamlined, efficient approach to data integration. The end user can upload data manually or use the REST API for the data transfer from the isolated server of the end users to the Knowledge Base. Figure 12 is an example of operator presence information parsed and converted into required JSON format which is the list of JSON objects.

```
{
    "ID": "100001",
    "Present": "True",
    "Timestamp": "11/20/2023 13:42"
},
{
    "ID": "100002",
    "Present": "False",
    "Timestamp": "11/20/2023 13:42"
},
```

Figure 12 Operators presence in JSON format.

The Knowledge Base provides support for uploading the data in CSV format as shown in below Figure 13.

ID	Present	Timestamp
100001	TRUE	11/20/2023 13:42
100002	FALSE	11/20/2023 13:42
100003	FALSE	11/20/2023 13:42
100004	TRUE	11/20/2023 13:30
100005	TRUE	11/20/2023 13:33
100006	FALSE	11/20/2023 13:33
100007	FALSE	11/20/2023 13:33
100008	FALSE	11/20/2023 13:33
100009	TRUE	11/20/2023 13:31
100010	TRUE	11/20/2023 13:40
100011	TRUE	11/20/2023 5:20

Figure 13 Operator presence in CSV format.

The below API methods from the Knowledge Base can be utilized to send the required data from the end use case site to knowledge base.

POST `/api/bkd/aggr/{repository}/{model}/{node}/{prop}/{token}` Append values to the aggregated property in JSON format

POST `/api/bkd/aggr_csv/{repository}/{model}/{node}/{prop}/{token}` Append values to the aggregated property in CSV format

Figure 14 Knowledge Base REST API endpoints.

4 SHOWCASE EXAMPLES

In this section will be presented the activities and methods that will be necessary to evaluate the tools produced during the project. The evaluation will concern two fundamental aspects. Firstly, the usability of the user interface, secondly the results produced by the decision model.

4.1 Human Machine Interaction Validation:

The focus of the HMI evaluation is its usability level. This approach will have as a target to investigate the “easy to use” design of the functionalities and identify potential issues preventing an inefficient and unsatisfactory interaction. In addition, the outcome of the test can provide guidelines and recommendations for potential changes.

A usability test consists of making a user group do typical tasks, using the software in a controlled environment. A sample of users is chosen that is representative of the category of users that will use the software. The users will be volunteer colleagues of the CRF, as the skills of the working group are very similar to those present in the line staff with decision-making tasks. In accordance with literature, a usability test with 6 - 9 users allow to capture 90% - 99% of the critical elements of interaction with a product.

Each user performs the same tasks separately. Their behavior is observed and analyzed to understand if, where and why they encountered difficulties. At the end of the test, participants will be given a questionnaire to collect some impressions on the product and the evaluated tasks.

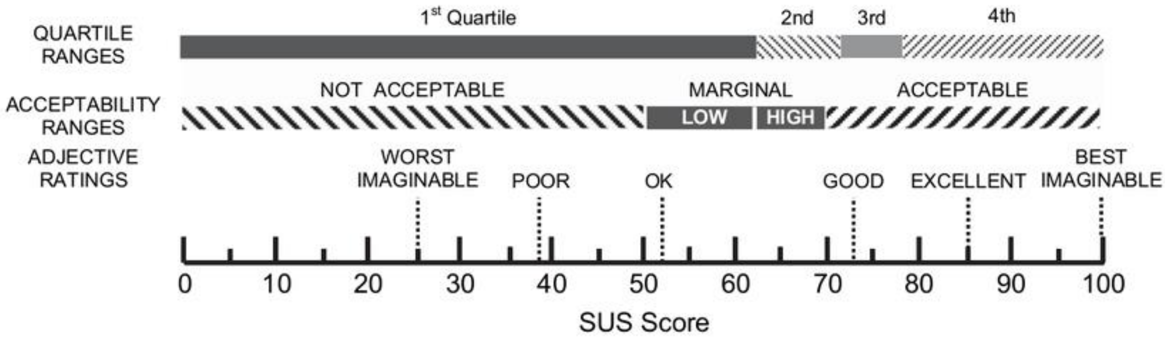
The ISO 13407 standard (adopted by FCA) identifies three fundamental types of assessment reports, depending if their purpose is to provide feedback for design, or to demonstrate compliance with specific standards, or to provide evidence of achieving human-centered goals (such as in terms of usability).

In particular, the software will be evaluated regarding the following criteria:

Make visible the invisible or highlight the most important information (bad inspections, etc.)

1. Maintain consistency
2. Provide informative feedback
3. Designing dialogs with closure
4. Offer an easy prevention and management of errors
5. Allow to fix errors (undo)
6. Indicate the possible correct alternatives
7. Allow a smooth turn of the actions
8. Make difficult the possibility of irreversible operations
9. Simplify the task structure
10. Reduce short-term memory load
    a. Keep the screens simple
    b. Provide online access to syntax rules, abbreviations, codes ...
    c. Do not create situations where users have to write a code on a spreadsheet to re-insert it later: if it has already been inserted the computer should know it.

To quantify the result of the questionnaires, the SUS system will be used. The System Usability Scale (SUS) provides a "quick and dirty", reliable tool for measuring the usability. It consists of a 10-item questionnaire with five response options for respondents; from Strongly agree to Strongly disagree. Originally created by John Brooke in 1986, it allows you to evaluate a wide variety of products and services, including hardware, software, mobile devices, websites and applications.



The usability test will be divided in 3 phases:

1. *Test preparation*: the evaluation team defines a series of task that the user must do (protocol definition) Than the team writes a questionnaire about the interface use.
2. *Actual test*: The test is done by every user applying the protocol, then the questionnaire is compiling. Normally the actions and the comments of the users are registered.
3. *Results analysis*: At the end of tests the evaluation team synthetize in a report the critical issues and the comments.

The interface is currently being defined. Based on the difficulty of using the interface, in the opinion of the technical partners, one or more training sessions and/or a written guide and/or a tutorial will be organized in advance.

4.2 Decision-making model Validation:

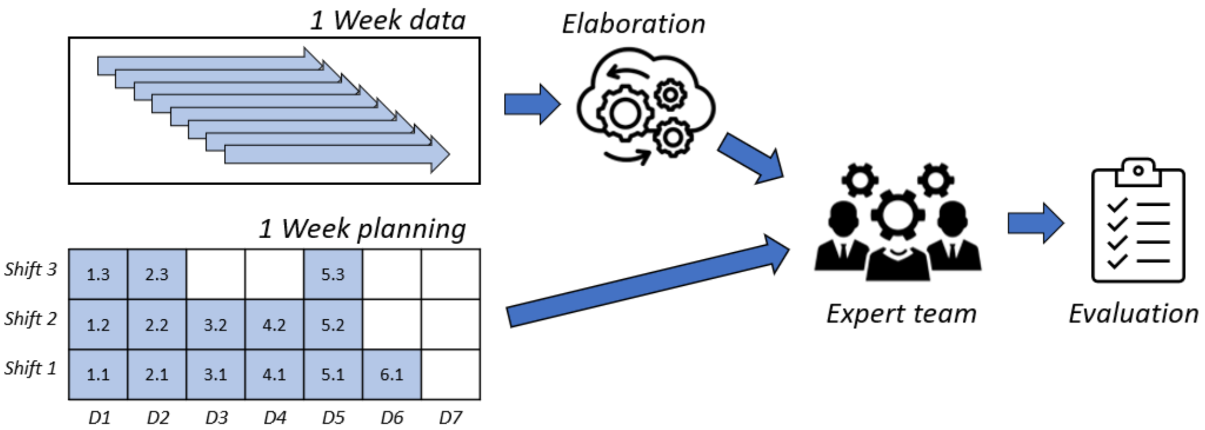
The second aspect that will be evaluated is the model's ability to make correct decisions. For this evaluation, volunteers will not be used based on their technical profile, but rather industry experts will be involved. In this case, in fact, the experience gained in the field is an essential parameter of choice.

The expert team will be composed by:

- Quality manager
- Logistics manager
- UTE Head
- Production planner

With reference to the first scenario (workload balance), the UTE head together with the production planer will provide the evaluation.

The test method will be to compare the decisions taken by the Pres shop staff with those proposed by the decision-making model. This will happen on the basis of the decisions already taken by the staff, which will be proposed to the system as new and therefore evaluated by the staff.



In other words:

- the planning, by shifts, of a week of work will be considered. At the same time, data useful for planning will be collected for the same week.
- The data for each shift will be synchronized one by one with the "FAIRWork DAI-DSS Database", generating alternative schedules for each shift.
- These plans will be analyzed by the team of experts who will provide an evaluation for each of them.

This process will be useful in a first phase to "calibrate the system" and highlight shortcomings or aspects not adequately taken into consideration. A continuous improvement process will then follow until a reliable model is obtained.

Once the model will be frozen, the evaluation of KPIs for the scenario will be carried out, described in section 2.2.4 of Deliverable D4.1.1. Regarding this aspect, the DECISION TIME and the ALTERNATIVE PLANS will be easy to evaluate. However, as regards the third KPI (UNPLANNED ABSENTEEISM), the evaluation can only be done on annual historical data, once the model will be actually used in Plant.

5 SUMMARY AND CONCLUSIONS

Section 2 of the deliverable describes the overall infrastructure set-up process at the use case partner side. Then, in section 3, we explain in more detail the technical preparation tasks such as data acquisition, human expert training, user selection and implantation of the current DAI-DSS prototype as a use case site and integration with the legacy systems. In principle, the project has made good progress in this preparation process with a slide delay based on a factory reorganization in Flex, this may result in an in adjustments for the use cases for Flex.

In principle the project made good progress with a slide delay based on Factory reorganization in Flex, this can result in adjustments for the use cases for Flex.

Section 4 describes the overall testing procedure. This includes a short description of the user evaluation and the corresponding KPIs for testing the use cases in general. It was clear identified that the DAI-DSS is a helpful tool for decision making in the future at Flex and CRF in the first iteration.

In the second iteration, we plan to add and integrate more functionality into the prototype. We expect further challenges as we plan to integrate it more deeply with some legacy

The challenge will be to integrate the tool directly into the company network so that direct and real-time data transfer is possible