AGILEHAND

D3.3 – AGILEHAND SUITEs Integration v1

WP3 – DESIGN: AGILEHAND Architecture and Integration

AGILEHAND has received the funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101092043.

Document Information

GRANT AGREEMENT NUMBER	101092043	ACRONYM	AGILEHAND
FULL TITLE	Smart grading, handling, and packaging solutions for soft and deformable products in agile and reconfigurable lines		
START DATE	01-01-2023	DURATION	36 months
PROJECT URL	https://agilehand.eu/		
DELIVERABLE	D3.3 – AGILEHAND SUITEs Integration v1		
WORK PACKAGE	WP3 – DESIGN: AGILEHAND Architecture and Integration		
DATE OF DELIVERY	CONTRACTUAL	December 2024	ACTUAL	December 2024
NATURE	R – Report	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	KBZ		
RESPONSIBLE AUTHOR	KBZ		
CONTRIBUTIONS FROM	UNI, UPM, UPV		
TARGET AUDIENCE	1) **AGILEHAND** Project partners;		
DELIVERABLE CONTEXT/ DEPENDENCIES	This document has no preceding documents and it is the first version of AGILEHAND SUITEs integration. A second version will be presented (D3.4) at the end of the project.		
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None		
READING NOTES	None		
ABSTRACT	This deliverable explains activities performed in WP3 specifically Tasks T3.3 and T3.4 with respect to the design and integration of solutions developed within the AGILEHAND project.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	24/09/2024	ToC	ToC created	KBZ
0.2	20/10/2024	Draft	Added Common Platform Sections	KBZ
0.3	30/10/2024	Draft	Added sections on SSO, Keycloak and GitLab deployment	KBZ, UNI
0.4	15/11/2024	Draft	Included pilot use cases and integration scenarios	KBZ
0.5	10/12/2024	Final	Finalized document ready for internal review	KBZ
0.7	19/12/2024	Review	Review and Implement suggestions	UPM, KBZ

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
AF	Architecture Framework
API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CR^{Hand}	Collaborative Robot Handling
DS^{Sense}	Data Sets Sensing
DT^{Agile}	Data Driven Digital Twin
ETA	Estimated Time of Arrival
FaaS	Functions as a Service
GQ^{sense}	Grade the Quality Sensing
IoT	Internet of Things

ICT	Information and Communication Technologies
IDS	International Data Spaces
KER	Key Exploitable Result
KPI	Key Performance Indicator
MES	Manufacturing Execution System
ML	Machine Learning
OER	Other Exploitable Result
OPC/UA	Open Platform Communications/Unified Architecture
PE^{Agile}	Production Execution Optimization Toolkit
PLC	Programmable Logic Controller
PR^{Agile}	Production Reconfiguration
PT^{Agile}	Product Oriented Traceability
RGB	Red Green Blue
RGBD	Depth (D) and Color (RGB)
ROS	Robot Operating System
RR^{Hand}	Robot Robot Handling
SC^{sense}	Self-Calibrating Sensing
SDKs	Software Development Kits
SMED	Single Minute Exchange of Dies
SQL	Structured Query Language
SSO	Single Sign-On
SUS	System Usability Scale
ST^{hand}	Self-Adaptable Transportation System
UI	User Interface
UX	User Experience
WP	Work Package
IAM	Identity and Access Management
RBAC	Role-Based Access Control
MFA	Multi-Factor Authentication
OAuth2	Open Authorization 2

SAML	Security Assertion Markup Language
CI/CD	Continuous Integration / Continuous Deployment
MRs	Merge Requests
SDLC	Software Development Lifecycle
VCS	Version Control System
SAST	Static Application Security Testing
DAST	Dynamic Application Security Testing
CVAT	Computer Vision Annotation Tool

Executive summary

This deliverable outlines the comprehensive activities from WP3 in the design, deployment, and integration of various solutions within the AGILEHAND project. It details the architectural framework, technical specifications, and integration strategies essential for creating a cohesive and scalable system tailored to agile and reconfigurable manufacturing environments. The document highlights the deployment of key components such as Single Sign-On (SSO), role-based access control, and modular frameworks, alongside the use of open-source tools like Keycloak for authentication, GitLab for source code management, and Superset for data visualization. Additionally, it includes pilot use cases, such as the Produmar project, demonstrating the practical applications and transformative potential of AGILEHAND solutions in real-world industrial contexts. The iterative development and validation of these solutions emphasize the project's focus on aligning technical architecture with stakeholder needs, ensuring modularity, interoperability, and user-centric design.

Document structure

Section 1. This section defines the technical specifications and requirements for AGILEHAND solutions, focusing on how they align with the needs of each pilot. The pilots—MULTISCAN, SANTORSOLA, PRODUMAR, and MARELEC—demonstrate practical applications of the solutions across different industrial contexts. Each suite is mapped to specific solution requirements, ensuring alignment with industrial needs. Solution diagrams and the importance of individual components are explored, emphasizing their roles in the AGILEHAND architecture. The section serves as a guide for developing the technical architecture by providing implementation maps and integration strategies for the identified components.

Section 2. This section details the design and testing of the AGILEHAND platform's user interface, focusing on usability and accessibility. Design tools such as Figma are used to create interactive mock-ups, while a co-creation approach ensures that the platform reflects end-user needs and feedback. Usability testing measures task success rates, error rates, and time-on-task to identify areas for improvement. The AGILEHAND common platform is tailored to different user roles, including admin, organization manager, and organization member. This ensures that the platform is both user-friendly and functional, supporting varied workflows and organizational requirements.

Section 3. This section outlines the integration of AGILEHAND solutions within a unified architecture. The architectural blueprint supports both standalone and cloud-based solutions, enabling seamless interoperability across the ecosystem. The common platform serves as a central interface for accessing multiple solutions, with Single Sign-On (SSO) enhancing user experience through simplified authentication. Data visualization tools, such as Superset, provide insights into key performance metrics, while GitLab ensures effective source code management and CI/CD pipelines streamline development workflows. Keycloak is utilized for role-based access management, ensuring secure and efficient user interactions within the platform.

Section 4. This section summarizes the contributions of the deliverable, emphasizing the iterative development of solutions and their validation across pilot scenarios. The focus is on aligning technical architecture with the needs of stakeholders, ensuring modularity, interoperability, and user-centric design. The deliverable establishes a strong foundation for implementing AGILEHAND solutions, addressing both technical and functional requirements in alignment with project objectives.

1. AGILEHAND Solutions Overview

This section provides a summary of the requirements derived from the Functional and Technical Specifications under T3.2, organized by suite and pilot, to guide the development of the AGILEHAND technical architecture. The primary goal is to develop the technical components required to integrate the AGILEHAND solutions, ensuring seamless interconnectivity and functionality across all systems.

The development process will involve the careful selection of technologies essential for the proper implementation of these solutions, building upon the inputs from the Functional and Technical Specifications and the activities outlined in the Usage Viewpoint. Implementation maps for associated components will also be provided to establish a clear and structured roadmap for deployment.

This technical architecture development process will provide a foundation for the implementation of AGILEHAND solutions, addressing the specific needs of the Smart Sensing Suite, Self-Adaptive Handling, Sorting, and Packaging Suite, and Agile Flexible and Rapid Reconfigurable Suite. The resulting architecture will empower the seamless integration and operation of solutions across diverse pilots, delivering a robust, scalable, and interoperable system tailored to the needs of the AGILEHAND project.

Solution Requirements and Specifications

SUITE: Smart Sensing Suite Requirements (Across All Pilots)

For Pilot 1 – MULTISCAN, the objective is to enhance the grading of soft and deformable products using advanced sensing technologies. The requirements include input from interconnected sensors, high-resolution data for quality assessment, real-time sensing data integration with existing systems, and modular sensing capabilities for easy upgrades.

For Pilot 2 – SANTORSOLA, the objective is to enable traceability of fresh products with advanced sensing systems. The requirements include the integration of sensors for monitoring environmental conditions, visual and numerical data acquisition for product grading, and enhanced precision in sensing deformable objects.

For Pilot 3 – PRODUMAR, the objective is to improve the quality control of seafood products using adaptive sensing. The requirements include underwater sensor deployment for real-time data acquisition, machine vision capabilities for defect detection, and data streaming to a central cloud for analysis.

For Pilot 4 – MARELEC, the objective is to achieve seamless grading of diverse food products. The requirements include multi-modal sensor fusion for simultaneous texture, color, and size detection, as well as integration with edge computing for real-time decision-making.

SUITE: Self-Adaptive Handling, Sorting, and Packaging Suite Requirements (Across All Pilots)

For Pilot 1 – MULTISCAN, the objective is to automate the sorting and handling of delicate items. The requirements include adaptive robotics for handling soft materials, real-time feedback systems for collision avoidance, and AI-based algorithms for dynamic sorting decisions.

For Pilot 2 – SANTORSOLA, the objective is to optimize packaging solutions for variable product sizes. The requirements include flexible packaging systems that adjust to different shapes, integration with vision systems for accurate sorting, and high-speed robotic manipulation for efficient throughput.

For Pilot 3 – PRODUMAR, the objective is to ensure minimal damage in handling and sorting seafood. The requirements include the use of soft robotic grippers, adaptive handling techniques based on object weight and fragility, and integration of predictive models for sorting anomalies.

For Pilot 4 – MARELEC, the objective is to streamline the packaging process with minimal human intervention. The requirements include automated sorting and packaging solutions, real-time data processing for accurate sorting decisions, and the use of modular robotic systems.

SUITE: Agile Flexible and Rapid Reconfigurable Suite Requirements (Across All Pilots)

For Pilot 1 – MULTISCAN, the objective is to enable quick reconfiguration of the grading system for new product types. The requirements include modular hardware for easy reconfiguration, software adaptability for integrating new grading algorithms, and scalable system architecture to accommodate diverse products.

For Pilot 2 – SANTORSOLA, the objective is to adapt production lines to seasonal product variability. The requirements include dynamic workflow adjustments based on product data, reconfigurable conveyor systems, and integration of AI-driven process optimization tools.

For Pilot 3 – PRODUMAR, the objective is to increase adaptability of processing seafood with varying characteristics. The requirements include modular designs for quick equipment swaps, cloud-based systems for centralized control, and real-time analytics for production optimization.

For Pilot 4 – MARELEC, the objective is to achieve seamless integration across production lines. The requirements include unified interfaces for controlling multiple solutions, real-time synchronization across sorting, grading, and packaging systems, and cloud-integrated reconfiguration for scalability.

2. AGILEHAND UI/UX

This section has been previously described in detail in Deliverable D3.2. Only an overview of actions performed are demonstrated here.

Design Tool and Methodology

Figma is a cloud-based design tool that supports a wide variety of functionalities essential for designing user interfaces. It is widely accepted in enterprise practices due to its comprehensive features, including real-time collaboration, which allows multiple designers to work on the same design simultaneously. Stakeholders can view and provide feedback directly on the design as it evolves.

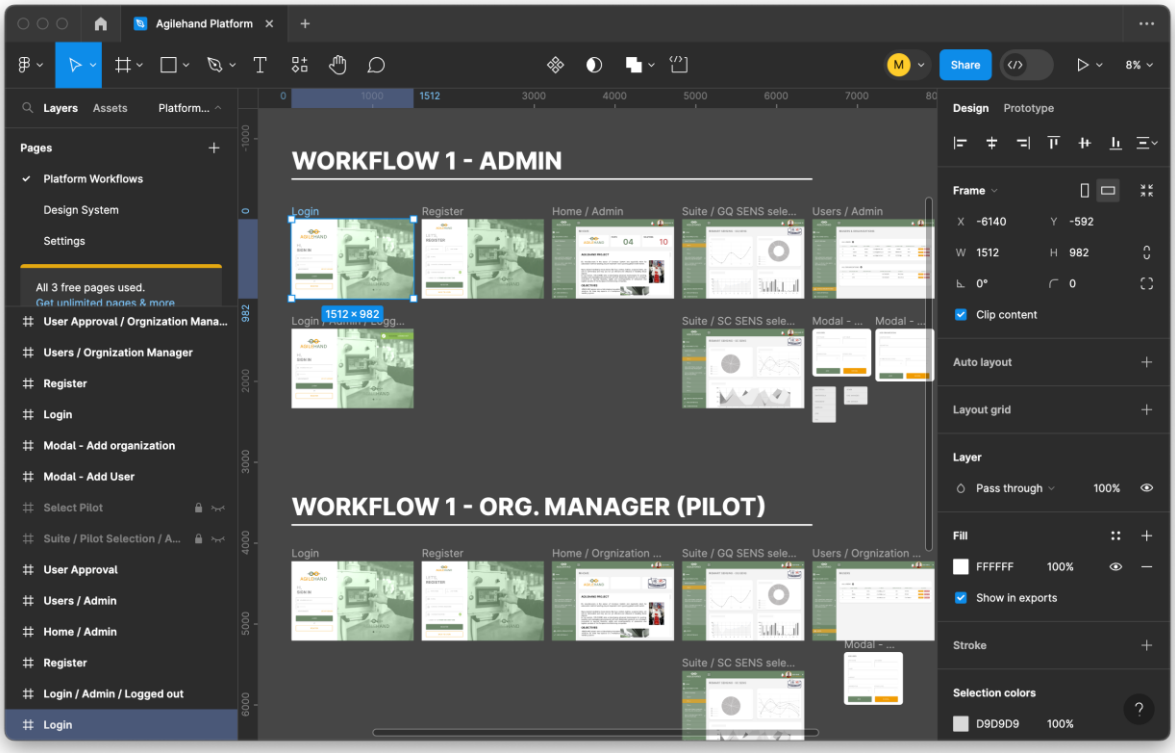

Figure 1 - Figma Editor

Figma offers robust prototyping capabilities to simulate application workflows. These prototypes provide screen-to-screen navigation, interactable animated components, and responsive design, ensuring a consistent user experience across multiple devices and screen sizes. This functionality is crucial for testing and refining user interfaces before development.

Co-Creation Approach

A high-quality user experience is largely dependent on the user-friendliness of the UI. To achieve this, the co-creation approach involves continuous collaboration and participation of end-users. This approach gathers feedback and insights from users, which are then used to refine the product design, ensuring it meets user expectations.

The AGILEHAND common platform mock-up is designed using co-creation approach. Active participation from end-users ensures that the final system is built on user expectations, making it more accessible, user-friendly, and highly interactive.

Mock-ups and Testing Framework

AGILEHAND Common Platform acts as a central dashboard offering analytics and insights from various AGILEHAND solutions. It accommodates distinct user types based on roles and permissions, each with unique workflows and requirements.

User Roles

Common Platform provides different user roles and each role have associated limitations based on functionalities provided.

Admin role is responsible for overall management of the platform, including managing solutions, adding and managing organizations, and their managers and any platform configuration.

Organization Manager role can manage associated organization and its members. Organizations can be assigned one or more solutions tailored to their needs, including pilot companies and other project partners.

Organization Member role uses the solutions provided by the organization with minimal dashboard features.

Pages

Common Platform web UI consists of following unique pages that allow management and configuration functionalities depending on the user role

Login Page

Registered users can log in with their email and password. The page also provides access to the registration page and a password recovery option.

Registration Page

New users can register by providing their information. They must be approved by the organization manager to gain access.

Homepage

Features an overview of the AGILEHAND project, its objectives, and the number of pilot studies and solutions. It provides streamlined access to the project's fundamental components.

User Approvals Page

Admins and organization managers can review and approve new user access, ensuring appropriate access rights.

Usability Testing

For conducting usability tests for AGILEHAND Common Platform UI/UX a standardized method was used which was based on metrics provided by the System Usability Scale (SUS).

System Usability Scale (SUS) is a widely adopted standard for evaluating user interface usability through quantitative metrics.

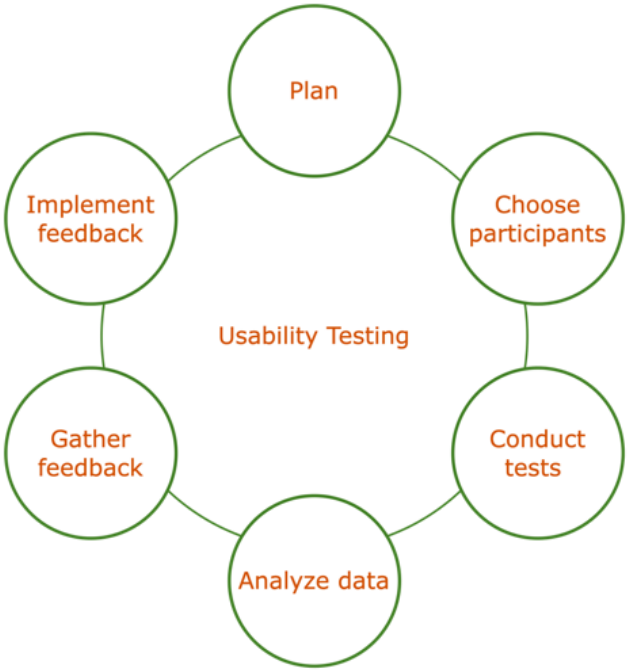

Figure 2 - Usability testing process

SUS consists of following metrics:

- **Task Success Rate**: The proportion of participants who completed given tasks. A higher rate indicates ease of use.
- **Time of Task**: Measures the time taken to complete each task, helping identify potential issues.
- **Error Rate**: Errors during tasks, such as misclicks or incomplete tasks.
- **Test Completion Time**: The overall time taken to complete the usability test.
- **Click Heatmaps and User Flow Patterns**: Tracking user clicks and navigation to identify common behaviours and optimize UX.
- **Questionnaires and Surveys**: Collecting qualitative feedback on UI perception, satisfaction, preferences, and suggestions.
- **Test Scenarios**: Participants perform tasks and provide feedback on the user interface. Tasks include navigating from the login page to the homepage, selecting solutions from the menu, opening and closing panels, and managing users and organizations.

Usability Testing Process

The usability assessment included a series of test scenarios with specific tasks for participants to perform. These tasks were designed to evaluate the ease of navigation and interaction with the platform. Following tasks were provided to the participants:

- **Task 1**: Navigate from the login page to the homepage.
- **Task 2**: Select the GQ SENS solution from the AGILEHAND Suites menu.
- **Task 3**: Select the SC SENS solution from the AGILEHAND Suites menu.
- **Task 4**: Open and close the notification panel.
- **Task 5**: Open and close the profile dropdown.
- **Task 6**: Navigate to the Users & Organizations page.
- **Task 7**: Open the Add User form.
- **Task 8**: Open dropdown options in the Add User form.
- **Task 9**: Open the Add Organization form.
- **Task 10**: Navigate to the User Approvals page.
- **Task 11**: Navigate back to the homepage.

After performing the tasks, the participants were asked to fill in a feedback questionnaire. The participants answered following questions to provide qualitative feedback on their experience:

- **Question 1**: On a scale of 1 to 5, how easy was it to navigate through the dashboard?
- **Question 2**: How long did it take you to complete all the tasks using the dashboard?
- **Question 3**: On a scale of 1 to 5, how many clicks were required to access different sections of the dashboard?
- **Question 4**: How satisfied are you with the overall design and layout of the dashboard?
- **Question 5**: Were there any specific elements or features that you found confusing?
- **Question 6**: Can you provide any suggestions for improving the overall user experience of the dashboard?

Usability Testing Platform

To execute the testing process a cloud-based usability testing platform called Useberry was used. Useberry is an online platform that allows usability testing for digital prototypes of web and mobile apps. It integrates with various design tools, including Figma.

Useberry collects rich insights and user feedback directly from the prototype through a streamlined workflow. It gathers data such as session recordings, click tracking, and user flows, providing comprehensive analysis for improving the design.

Findings, Results Analysis, and Recommendations

Usability testing provided valuable insights into user behaviours and areas for improvement. **Heatmaps and User Flows** identified common behaviours and areas needing optimization. For example, user click maps on the login and home pages revealed parts of the design that could be improved for better interaction. **Completion Rates and Times** highlighted tasks that were easy or

difficult for users. For instance, Task 1 had a high completion rate, indicating it was easy to accomplish, while Tasks 2 and 10 had lower completion rates, suggesting potential issues. Feedback questionnaire provided useful **Recommendations** to enhance the overall user experience.

A total of 8 AGILEHAND partners participated in the usability testing. Based on the findings and recommendations, changes were made by addressing design issues and improving UI flexibility and interaction.

3. AGILEHAND Solutions Integration

Architectural Overview

The AGILEHAND platform is designed with a robust architecture that supports flexibility, interoperability, and scalability, catering to the unique demands of agile and reconfigurable manufacturing environments. This section provides an in-depth look at the platform's functional and technical specifications, detailing the key components and design principles that enable seamless integration of various AGILEHAND suites.

AGILEHAND's architecture is built to bridge the gap between standalone and cloud-based solutions, offering centralized management while maintaining the autonomy of each suite within the ecosystem. Core features, such as the Single Sign-On (SSO) system, user role management, and a modular framework, contribute to an integrated, user-friendly experience that aligns with industry standards for data security, accessibility, and system flexibility. The functional design emphasizes interoperability across the SMART SENSING, SELF-ADAPTIVE HANDLING, and AGILE RECONFIGURABLE suites, enabling real-time data exchange and synchronized workflows to enhance production line efficiency.

The technical specifications of AGILEHAND further enhance its adaptability, with cloud-native capabilities and advanced analytics tools that support monitoring, visualization, and control of manufacturing processes. The use of AI-driven solutions allows the system to adjust dynamically to changes in product types, production volumes, and quality requirements, thus reducing downtime and optimizing resource allocation. This section delves into these specifications, showcasing how AGILEHAND's design enables manufacturers to meet the evolving demands of modern production systems.

The architectural design of the AGILEHAND platform is built to support the diverse and demanding needs of modern manufacturing environments that require flexibility, agility, and integration across a wide range of processes. The AGILEHAND architecture employs a cloud-native, modular approach that enables both centralized management and decentralized deployment, accommodating the specific needs of each solution within the ecosystem while ensuring seamless interoperability.

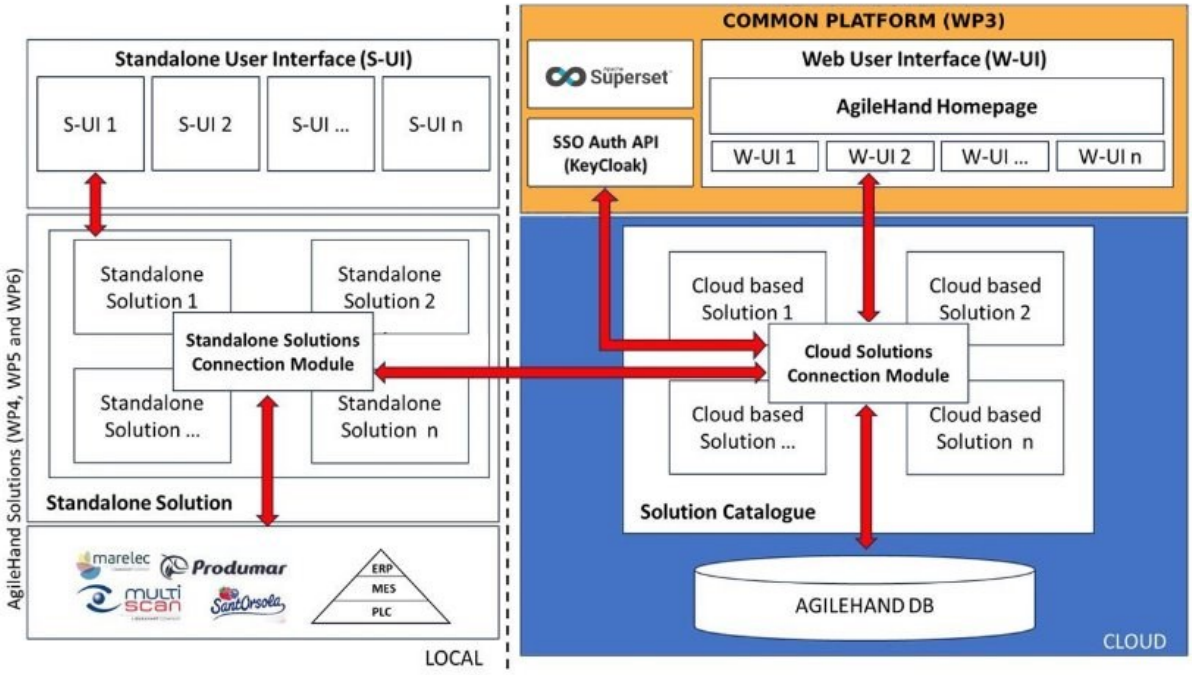

Figure 3 - AGILEHAND architectural overview

The architecture highlighted in Figure 3 shows key components of the architectural blueprint of the AGILEHAND ecosystem. Leveraging Keycloak, a robust open-source SSO and identity management solution, AGILEHAND provides a secure authentication and authorization framework. This allows users to access all AGILEHAND solutions under a unified login, which simplifies user management and enhances data security. Different user roles and permissions are configured, ensuring that each user's access is tailored to their specific needs and responsibilities within the platform.

Each AGILEHAND suite is assigned a dedicated space within the platform. These modular areas ensure that each suite operates autonomously yet can easily interact with other suites through a standardized data exchange protocol. This design allows individual solutions to be updated, maintained, or scaled without affecting the overall functionality of the platform.

The platform integrates Apache Superset as a data visualization tool to provide real-time insights across all AGILEHAND solutions. This tool connects with the platform's data sources, enabling users to visualize key performance indicators (KPIs) and production metrics through customizable charts and dashboards. This visualization layer allows for both high-level monitoring and in-depth analysis, making it easier for users to make informed decisions and quickly address any performance issues.

AGILEHAND Common Platform

All the solutions developed in AGILEHAND ecosystem are standalone except some which depend on each other. Since an aggregate of many solutions is utilized in a pilot it is necessary to provide an interface where multiple solutions are accessible under one common platform. The solutions are divided in two groups based on their availability on cloud and a web-based user interface. For the solutions that are cloud based and provide a user-interface for interaction are grouped into integration for cloud-based solutions and the solutions that work with other solutions or standalone but do not provide a user interface by default are categorized into Integration for

stand-alone solution. Common platform provides a common place to access all these solutions for any partner pre authenticated with proper role and permissions. The platform interface provides single sign on functionality to authenticate just one time, this authentication works across the ecosystem AGILEHAND apps.

The Common Platform provides a unified entry point to all solutions in the project. Each solution is assigned a dedicated space in the common platform. Solutions that are not directly interactive are visualized using open-source Apache Superset data visualization tool. Remaining solutions are integrated with a tailored user interfaces to facilitate interaction.

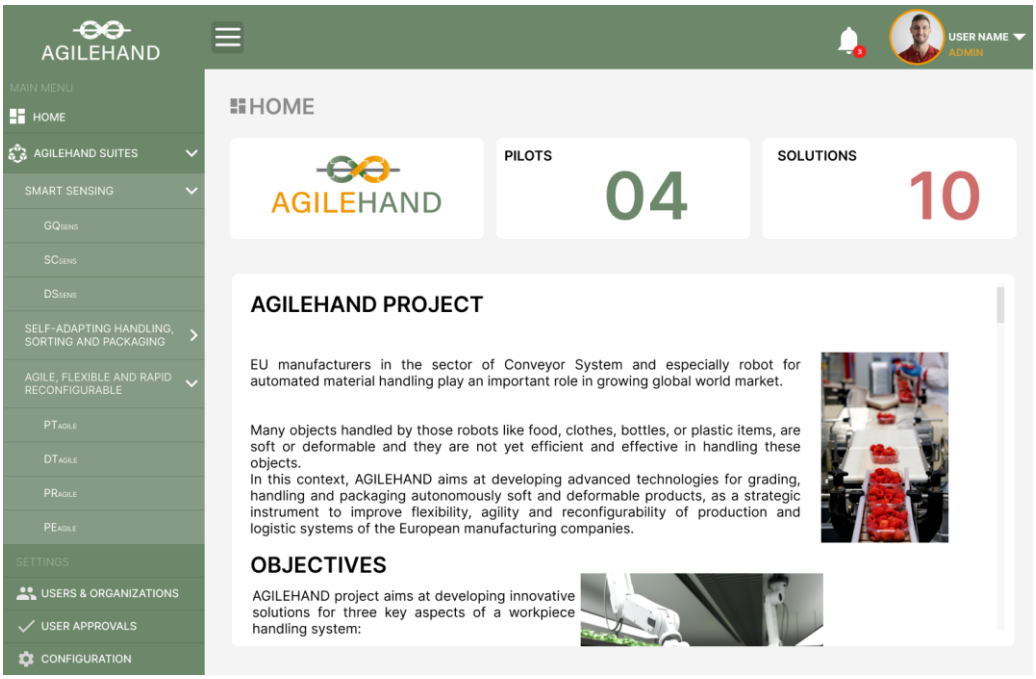

Figure 4 - Common Platform web UI homepage

The Common Platform is designed to serve as a unified interface within the AGILEHAND ecosystem, allowing users to access multiple solutions seamlessly. It integrates both cloud-based and standalone solutions, ensuring that users can interact with various tools and applications through a single, cohesive platform. This integration is facilitated by a single sign-on (SSO) system, which simplifies the authentication process and enhances user experience by requiring only one login to access all available solutions.

Features

The platform boasts several key features aimed at enhancing usability and security. The Single Sign-On (SSO) feature allows users to authenticate once and gain access to all integrated solutions, streamlining the login process. Additionally, the platform includes robust role and permission management, ensuring that access to different solutions and data is controlled based on user roles. Another significant feature is the KPI visualization tool, which provides users with the ability to visualize key performance indicators and metrics, aiding in data-driven decision-making.

Access Types

The Common Platform supports two main types of access: cloud-based solutions and standalone solutions. Cloud-based solutions, which come with their own dedicated user interfaces, are embedded directly into the platform. These solutions benefit from the platform's SSO and role-based access controls. Standalone solutions, which typically lack a default user interface, are integrated into the platform primarily for KPI visualization. This ensures that even solutions without direct user interaction can still provide valuable insights through the platform.

Admin: Admins are responsible for the overall management of the platform. Their duties include managing solutions, adding and managing organizations, and overseeing organization managers. Admins have the highest level of access and control, ensuring the smooth operation and integration of various solutions within the platform.

Organization Manager: Organization Managers handle the management of their respective organizations and its members. They are responsible for assigning solutions tailored to their organization's needs, which may include pilot companies and other project partners. Organization Managers ensure that their organization effectively utilizes the solutions provided by the platform.

Organization Member: Organization Members are the end-users who utilize the solutions assigned to their organization. They have access to the solutions with minimal dashboard features, focusing on using the tools and data provided to them. Their role is primarily operational, leveraging the solutions to perform their tasks and contribute to their organization's goals.

Configurations

The platform's configuration capabilities are centred around its integration with Superset. This integration allows the platform to connect with various KPI datasets, enabling comprehensive data visualization. Users can manipulate and display their data in multiple formats, such as charts and graphs, to gain deeper insights into their performance metrics. The configuration process ensures that each solution's data is accurately represented and easily accessible through the platform.

Organization Management

The platform's configuration provides management functions for a user with organization manager role to manage their users within the organization. They can add or remove users as required and can also manage their access to certain functionalities such as restricting access or providing view only access to certain solutions. The organization

Offline Ingestion

Supported KPI dashboards would allow users with configured access to upload structured data files such as in JSON format to directly view the results in the common platform without the need to fetch data online. The offline ingestion does not persist any data and the dashboard refreshes once the page has been reloaded. The offline ingestion is useful for quickly visualizing a batch of data from a short period without needing to feed the data to the KPI databases in the backend.

Visualizations

Superset

One of the standout features of the AGILEHAND Common platform is the Solution KPI Dashboard. This dashboard leverages an instance of Superset, a powerful data visualization tool, to display KPI data. Partners can configure and customize their dashboards to suit their specific needs, creating various charts and graphs to represent their data. These visualizations are then accessible through the common platform, providing a centralized location for monitoring and analysing performance metrics.

Deployment of Superset

The deployment of Superset within the AGILEHAND Project involves several key components, each playing a crucial role in ensuring secure and efficient data visualization and user authentication. Superset is deployed using Docker, which simplifies the setup and management of the application. Docker containers ensure that Superset runs consistently across different environments, providing a reliable platform for data visualization. Alongside Superset, a Postgres database container is deployed for demonstration purposes. This database stores Superset's metadata, including dashboard configurations, user data, and other essential information. The use of Docker for the Postgres database ensures easy deployment and scalability.

Superset communicates with a Keycloak instance to authenticate users. Keycloak provides a robust identity and access management solution, enabling Single Sign-On (SSO) across the AGILEHAND ecosystem. This integration ensures that users can log in to Superset using their existing credentials, enhancing security and user convenience. Additionally, the entire service, including Superset and the Postgres database, is protected using Cloudflare. Cloudflare provides security features such as DDoS protection, SSL/TLS encryption, and a Web Application Firewall (WAF), ensuring that the deployment is secure from external threats.

The deployment architecture leverages Docker containers for both Superset and the Postgres database. This approach provides several benefits, including isolation, portability, and ease of management. Docker Compose can be used to define and run multi-container Docker applications, simplifying the deployment process. Superset interacts with the Postgres database via TCP to store and retrieve metadata, ensuring that all configuration and user data is persisted, allowing for reliable and consistent operation.

When users attempt to log in to Superset, the authentication request is forwarded to the Keycloak instance. Keycloak verifies the user's credentials and, upon successful authentication, grants access to Superset. This integration leverages Keycloak's SSO capabilities, allowing users to authenticate once and access multiple services within the AGILEHAND ecosystem. Cloudflare sits in front of the Superset and Postgres services, providing a layer of security and performance optimization. Cloudflare's features include DDoS protection, SSL/TLS encryption, and a Web Application Firewall (WAF), which help protect the deployment from various cyber threats and ensure secure communication between users and the services.

The workflow begins with an external user sending a request to access Superset. The request first passes through Cloudflare, which provides security and performance enhancements. The request

is then forwarded to the Keycloak instance for authentication. Keycloak verifies the user's credentials and, if valid, grants access. The authenticated request reaches the Superset instance, allowing the user to interact with the dashboards and visualizations. Superset interacts with the Postgres database to retrieve and store metadata, ensuring that all user configurations and data visualizations are properly managed.

This deployment architecture ensures a secure, scalable, and efficient setup for Superset as a service for Common Platform, leveraging Docker for containerization, Keycloak for authentication, and Cloudflare for security.

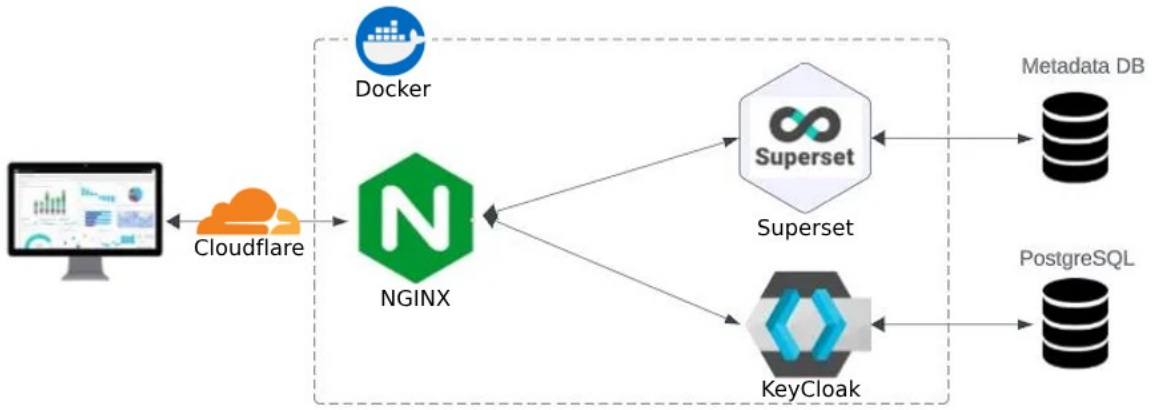

Figure 5 - Deployment of superset using docker

Data Integration

Superset integrates seamlessly with various data sources to provide comprehensive data visualization capabilities. It connects to the KPI datasets of different solutions within the AGILEHAND ecosystem, allowing users to create and customize dashboards that display key performance indicators and metrics. This integration ensures that users can manipulate and display their data in multiple formats, such as charts and graphs, providing valuable insights into their operations.

Supported Databases

Superset supports a wide range of databases, making it a versatile tool for data visualization. In the AGILEHAND Project, Superset is configured to work with a Postgres database for demonstration purposes. However, it can also connect to other popular databases such as MySQL, Oracle, and SQL Server, among others. This flexibility allows users to integrate Superset with their preferred data storage solutions, ensuring that they can leverage existing infrastructure while benefiting from Superset's powerful visualization capabilities.

Integration Scenarios

Integration of cloud-based solutions

The cloud-based solutions that are developed with their dedicated user-interface are embedded into the common platform as is but additional features of common platform such as control on access roles and permissions provided by the integrated SSO authentication system.

The solutions integrated using this approach are listed in navigation bar of AGILEHAND Common Platform based on the availability for the partner that is logged into the platform at that time.

This approach comes with additional requirements and efforts from the partners that are developing the solution such as the solution page is totally managed by the partner providing the solution and the communication with authentication API is integrated into the solution.

Components

In this scenario the Common platform works with three main components to provide centralized access and integration of solutions that have web frontend available in the cloud.

1. **WebUI:** Common Platform offers a unified WebUI to authenticated end users for accessing and viewing all user interfaces of different solutions in a restricted single pilot setting.
2. **KeyCloak Authentication API:** This API is responsible for doing the communication with Keycloak instance in the backend to authenticate users for both the cloud solutions and the common platform. Using this api the users only need to be authenticated one time at any moment in the process and Single Sign On would keep them logged in across the AGILEHAND ecosystem for the session.
3. **Solution UIs:** These are the barebone frontends provided by the solution developers to include only the components of the specific solution and the rest of the user interface such as sidebar, headers, menus are all provided by the Common Platform integration. The solutions UI also manages all the connections and data fetching from the pilot facility for the purpose of KPI visualization.

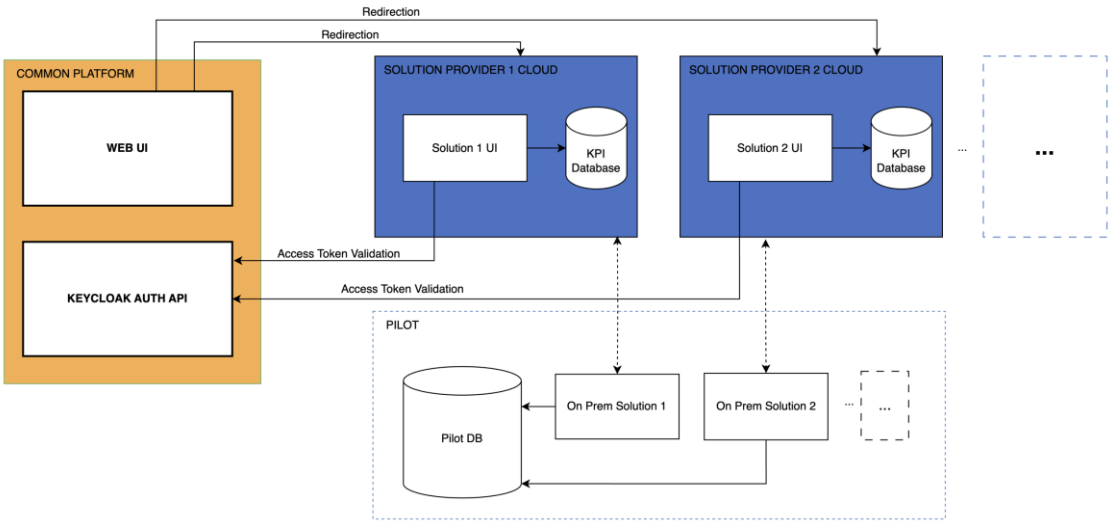

Figure 6 - Architecure for Integation of cloud-based solutions

Integration of stand-alone solutions

The integration of stand-alone solutions that by default are not developed with user-interface or lack user interaction are integrated using second approach. The difference in this approach from the previous one is that the common platform provides a way to visualize KPI data and metrices generated by these solutions. The visualization is provided by an instance of superset configured particularly for this purpose and integrated with the AGILEHAND SSO authentication system. This instance of superset provides connection to the solution's KPI dataset and visualization of many kinds to manipulate KPI data into various form of charts and graphs. This superset instance is provided in with the access to the common platform and lets partners configure dashboard for their solutions. Once the dashboards are created, they can be displayed in the solution specific section in the common platform web user interface.

Components

The integration of stand-alone solutions is achieved through a different approach, due to the lack user interfaces or direct user interaction. This method focuses on visualizing KPI data and metrics generated by these solutions. This approach introduces another component called Superset.

1. **WebUI**: Same as previous integration scenario, the WebUI also provides integration of preconfigured superset dashboards by embedding them in the solution specific areas. These dashboards can be accessed by authenticated end users based on the permissions and roles defined in keycloak and access type configured in the common platform, enabling them to access and view the user interfaces of various solutions within a controlled, single-pilot environment. This centralized interface simplifies navigation and enhances user experience by consolidating multiple solutions into one cohesive platform.
2. **KeyCloak Authentication API**: The authentication API is used in the same as described in the first integration scenario of cloud based solutions.
3. **Superset Instance**: For stand-alone solutions, the Common Platform integrates an instance of Superset specifically configured for visualizing KPI data and metrics. This Superset instance is connected to the AGILEHAND SSO authentication system, ensuring secure access. It allows users to connect to the KPI datasets of various solutions and create diverse visualizations, such as charts and graphs. Partners can configure dashboards within Superset, which are then displayed in the solution-specific sections of the Common Platform WebUI. This approach provides a powerful and flexible way to monitor and analyze performance data from stand-alone solutions, enhancing the overall value and usability of the Common Platform.

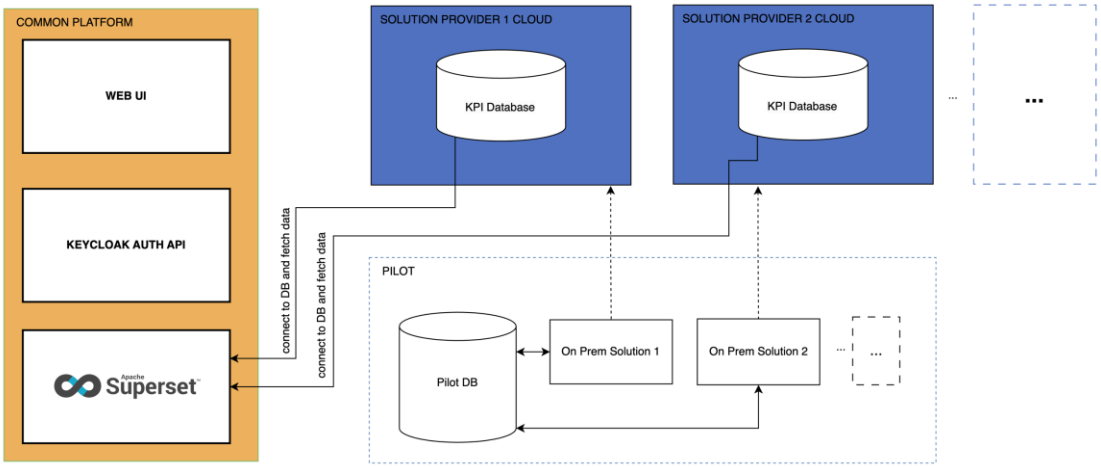

Figure 7 - Architecure for Integation of standalone solutions

Pilot Use Cases for Common Platform

The Produmar pilot project demonstrates the AGILEHAND platform's transformative capabilities in automating complex handling, sorting, and packaging processes for delicate and variable products. Produmar, a leader in the frozen fish processing industry, faces challenges typical of this sector, including the need for accurate, high-quality sorting and grading of fish products, currently handled manually by trained operators. The AGILEHAND solutions, particularly the SMART SENSING and SELF-ADAPTIVE HANDLING suites, are tailored to address these challenges by introducing automation to improve efficiency, reduce labor demands, and enhance product quality.

Before integrating AGILEHAND technology, Produmar's sorting process involved manual separation of mixed fish groups based on species, maturity, and quality attributes like texture, color, and size. Skilled labor is required to achieve the precision needed for this complex sorting and grading process, which is both time-consuming and prone to inconsistencies. Furthermore, the manual classification of fish fillets into quality categories (e.g., A-class and B-class) introduces variability, affecting product consistency and increasing operational costs.

An experimental setup has been setup to evaluate the effectiveness AGILEHAND Solutions in frozen fish industry. The data obtained from Produmar's facility included images for frozen fish steaks and fillets of different quality. Data acquisition utilized Intel RealSense D456 sensor, capturing RGB Depth images. The annotation of data was performed using well-known CVAT (Computer Vision Annotation Tool). For the first task (fish slicing), a total of 617 fish samples were acquired, while for the steak grading task, 103 examples were acquired. For both scenarios, the datasets were split into 80% for training and 20% to test the trained instance segmentation model. It is to be noted that for the use cases that consider the adoption of Intel® RealSense™, the images were always acquired with the highest resolution (1280x720) to allow the highest possible quality. For the same reason, the sensor was placed at the minimal distance allowed to enable depth measurement (i.e., about 60 cm).

The primary objective of the AGILEHAND pilot at Produmar is to implement semi-automated mechanisms for fish sorting and quality detection. By leveraging intelligent manipulators with self-adaptive capabilities, AGILEHAND enables automated sorting, grading, and handling

processes tailored to the physical properties of each fish, such as weight, size, and deformability. This approach aims to reduce the reliance on manual labor, improve sorting precision, and increase throughput.

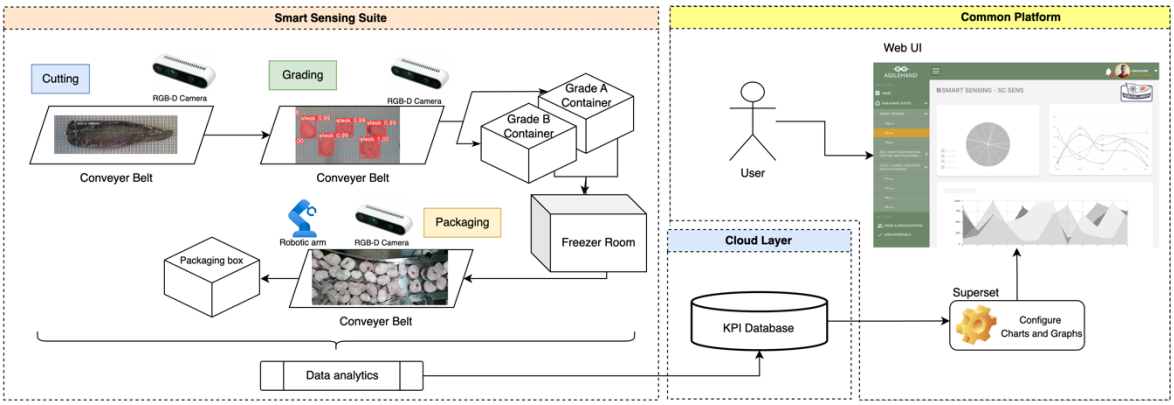

Figure 8 - Smart Sensing Suite example for Produmar

The AGILEHAND SMART SENSING suite provides real-time data on fish quality, guiding the cutting, grading, and sorting processes through sensor-based decision-making. Laser-guided indicators, for instance, help determine optimal cut points on each fish, ensuring consistency in product quality while reducing waste. Once the fish are cut into fillets, the system's sensors grade the fillets into predefined quality classes, automating a task that previously required extensive manual intervention.

The SELF-ADAPTIVE HANDLING suite utilizes robotic manipulators that handle fish fillets with precision and care, minimizing product damage during the sorting and packaging stages. This suite includes adaptive controls that adjust handling based on real-time sensory feedback, ensuring that each fish fillet is directed to the appropriate packaging stream according to its quality classification. The suite's collaborative capabilities allow it to work alongside human operators when necessary, creating a flexible, safe, and efficient work environment

Data gathered from AGILEHAND's sensors and robotic systems is processed and stored in a cloud-based database. This data, accessible through the AGILEHAND Common Platform, includes key performance indicators (KPIs) like product quality, sorting accuracy, and system throughput. These KPIs are visualized in real-time, allowing operators and management to monitor system performance, identify bottlenecks, and make data-driven adjustments to optimize production processes. The use of Apache Superset for visualization ensures that insights from AGILEHAND's systems are accessible, actionable, and aligned with Produmar's operational goals. The implementation of AGILEHAND's solutions at Produmar is expected to yield significant operational improvements, including:

- Enhanced Product Quality: The automated grading and sorting reduce human error, ensuring that each fish fillet meets consistent quality standards.
- Increased Efficiency: Automation reduces manual handling time, accelerating the sorting and packaging process and increasing overall throughput.

- Reduced Operational Costs: By minimizing reliance on skilled labor and improving resource allocation, Produmar can achieve cost savings in labor and waste reduction.

Improved Workplace Safety: Reducing the need for manual handling in cold and repetitive tasks lowers the risk of worker injuries and improves employee well-being.

The AGILEHAND platform is designed with scalability in mind, employing an agnostic and adaptive approach for UI/UX to ensure seamless user experiences in different environments and settings. Robust visualization capabilities are provided by integrating open-source technologies like Apache Superset that can be easily extended and customized as needed. Moreover, A standardized interface allows for smooth integration of both standalone and cloud-based solutions which ensures that system can be scaled efficiently and adapt to the evolving needs of diverse manufacturing environments.

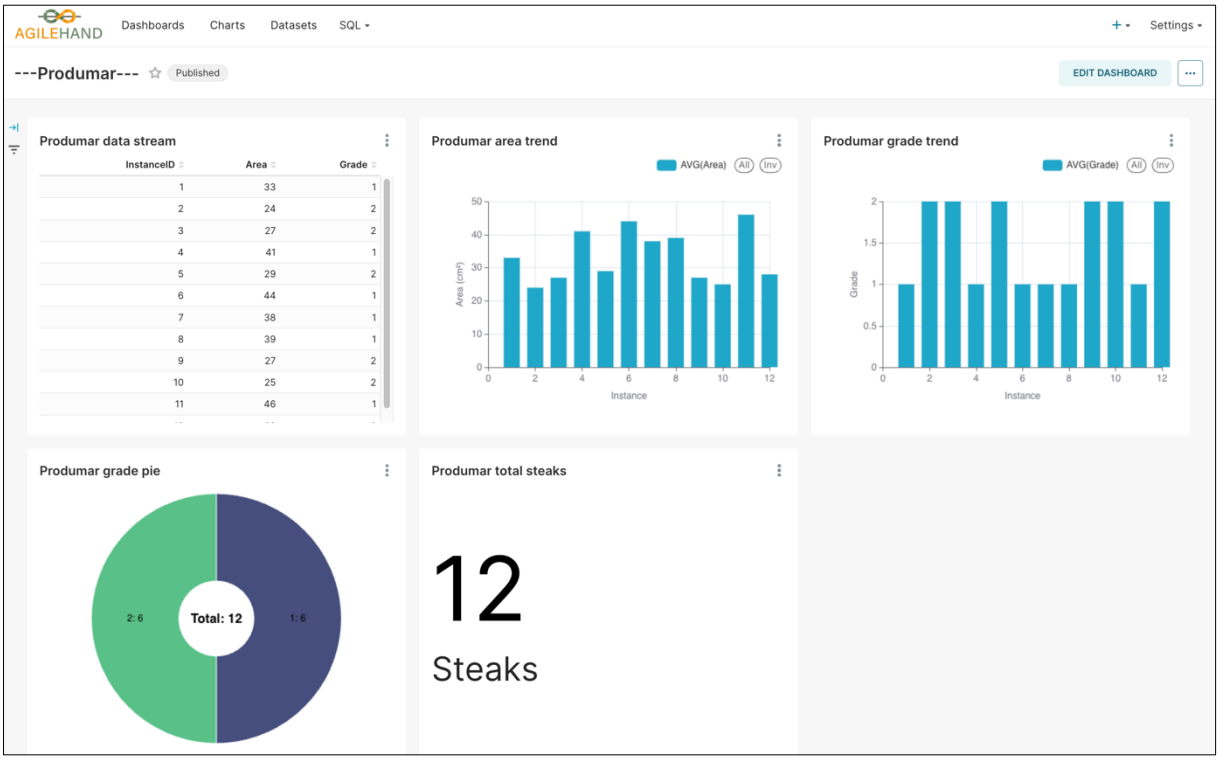

Figure 9 - KPI Dashboard for Produmar

Single Sign On (SSO) and Keycloak

Importance of SSO

Single Sign-On (SSO) is essential for organizations, offering key benefits across security, user experience, and IT management. By centralizing authentication, SSO enables users to access multiple applications with a single login, reducing password fatigue and improving productivity.

SSO streamlines access for users, requiring only one set of credentials for multiple applications. This not only reduces the frustration of managing multiple passwords but also allows users to

access resources more quickly, boosting productivity and reducing time spent on logins. Security is enhanced through centralized authentication, where organizations can enforce robust policies like multi-factor authentication (MFA) at one point of entry. Reducing the need for multiple passwords helps prevent weak password practices and lowers the likelihood of phishing attacks, as users encounter fewer login prompts and are more vigilant.

For IT administrators, SSO simplifies user management. Access rights can be adjusted centrally, allowing quick onboarding and offboarding of employees. This efficient management reduces help desk requests for password resets, cutting down on operational costs and improving support efficiency. SSO supports scalability by easing the integration of new applications. It allows companies to expand their IT stack without adding user complexity, providing a seamless experience as organizations grow and adopt new technologies.

Compliance and Auditing

SSO aids in compliance by logging user access data in a centralized system. These logs support regulatory compliance (e.g., GDPR, HIPAA) and provide an audit trail for investigating security incidents.

SSO is critical in modern enterprise environments for improving user experience, bolstering security, enhancing IT management, supporting scalability, and ensuring compliance. It allows organizations to operate securely and efficiently while enhancing both user and administrator experiences.

Keycloak

Keycloak is an open-source identity and access management (IAM) solution that provides single sign-on (SSO), user identity management, and secure access for applications. It supports a range of features tailored to both modern and legacy systems, making it a popular choice for centralized identity management. Keycloak allows users to authenticate once for multiple applications, improving user experience and efficiency.

User Identity Management

Keycloak integrates with identity stores such as LDAP and Active Directory. It supports user self-service options and role-based access control (RBAC) for assigning permissions, allowing fine-grained access control across applications.

Multi-Factor Authentication (MFA)

To enhance security, Keycloak includes MFA options that require users to provide additional forms of authentication, such as a one-time code along with a password, helping prevent unauthorized access.

Modern Standards: OAuth2, OpenID Connect, and SAML

With support for OAuth2, OpenID Connect, and SAML 2.0, Keycloak provides compatibility with a wide range of applications and enables secure token-based authentication, ideal for modern microservices and APIs.

Authorization Services

Keycloak's authorization services offer detailed access control, enabling administrators to create complex policies based on user attributes and conditions to enforce rules across applications.

Customizability and Extensibility

Keycloak allows customization of user interfaces, workflows, and plugins, making it adaptable to specific organizational needs and ensuring a branded experience.

Centralized Administration

The Keycloak admin console provides a web-based platform for managing users, roles, clients, and identity providers, simplifying user management and system configuration.

Cross-Platform, Cloud Ready, and Open Source

Cloud-native and scalable, Keycloak can be deployed on platforms like Kubernetes, supporting large-scale organizations. As an open-source solution under the Apache License, it's backed by a strong community and supported by Red Hat.

User management

User Identity Management (UIM), often integrated with Single Sign-On (SSO), is essential for secure and efficient IT operations in organizations. UIM centralizes user identities, simplifies access, and ensures security and compliance.

Centralized Identity Management

UIM consolidates identity profiles across systems, simplifying access management and allowing quick, secure onboarding and offboarding. This centralization is crucial for security, ensuring access is promptly updated as employees join or leave.

Enhanced Security

UIM enforces role-based access, ensuring users have the minimum necessary permissions. By integrating Multi-Factor Authentication (MFA), UIM enhances security beyond passwords, reducing identity spoofing risks. Centralized monitoring further enables quick anomaly detection.

Simplified Access and SSO Integration

UIM creates a central source of identity data, eliminating redundancies. When paired with SSO, UIM provides a seamless login experience and consistent security across applications, making policy enforcement straightforward.

Roles and permissions

Roles and permissions are essential for managing access control within Keycloak, enabling organizations to define and enforce user actions and resource access across applications and services.

Roles in Keycloak

In Keycloak, a role is a label assigned to users or groups that defines what actions they are allowed to perform. Roles are often used to group users with similar permissions. There are two main types of roles: **Realm Roles**, which are global across an entire Keycloak realm and apply to all

clients within that realm, and **Client Roles**, which are specific to individual applications and provide more granular control within each application.

Keycloak also supports **Role Mapping**, where roles can be assigned to users, groups, or clients. Roles can be directly assigned to users or inherited through group membership. Additionally, **Role Hierarchies** enable roles to inherit permissions from others, allowing for flexible, layered access control structures.

Permissions in Keycloak

Permissions in Keycloak allow administrators to control user access to specific resources or actions based on assigned roles. Keycloak's **Authorization Services** enable fine-grained access control through resource-based, scope-based, and policy-based permissions. **Resource-Based Permissions** define which roles or users can access specific resources, while **Scope-Based Permissions** specify which actions (e.g., read, write) can be performed on a resource. **Policy-Based Permissions** allow for complex access conditions, including time-based or attribute-based restrictions.

Permissions are categorized as **Resource Permissions**, controlling access to particular resources, and **Scope Permissions**, which define the specific actions allowed on those resources. Permissions are managed through Keycloak's Authorization tab, where administrators can assign resource policies and configure access control.

Groups and Roles

In Keycloak, groups can be assigned roles, allowing users within a group to inherit the group's permissions. This feature facilitates centralized role management for large numbers of users, allowing roles to be flexibly assigned through group membership.

Source Code Management

Gitlab Instance

GitLab is a comprehensive DevOps platform that provides tools for **version control**, **CI/CD (Continuous Integration/Continuous Delivery)**, **project management**, and **collaboration** for software development teams. It's an open-source, web-based tool built around Git, the widely used distributed version control system. GitLab also integrates with various development processes to offer a complete solution for managing the entire DevOps lifecycle.

Version Control

Git-based Version Control: GitLab uses Git as its underlying version control system, allowing users to track changes in their codebase, collaborate on code, and manage multiple branches for features, bug fixes, or releases.

Repositories: Each project in GitLab has its own repository where all versions of the code are stored. Users can clone the repository, make changes locally, and push the changes back to the remote GitLab repository.

Continuous Integration / Continuous Deployment (CI/CD)

- **GitLab CI/CD Pipelines:** GitLab's built-in CI/CD tool automates the testing, building, and deployment of applications.
- **Continuous Integration (CI):** Automatically runs tests on new code pushed to the repository, ensuring code quality before it's merged.
- **Continuous Deployment (CD):** Automates the process of deploying code to production environments once the code passes all tests and checks.
- **Runner System:** GitLab uses **runners** (agents that execute jobs defined in pipelines) to run CI/CD jobs. Runners can be either shared across multiple projects or dedicated to specific projects.

Project Management and Collaboration

- **Issue Tracking:** GitLab has an integrated issue tracking system that helps teams manage feature requests, bugs, and tasks related to the project. Each issue can be assigned, labeled, and prioritized.
- **Milestones and Roadmaps:** GitLab allows teams to create **milestones** and **roadmaps** to plan and track progress across multiple issues and projects, ensuring that all development efforts align with business goals.
- **Merge Requests (MRs):** Similar to **pull requests** in GitHub, GitLab uses **merge requests** to propose changes to a repository. Merge requests allow teams to review code, discuss changes, and integrate approved changes into the main branch.
- **Wikis and Documentation:** Each project in GitLab has an integrated wiki, which teams can use to document important project details, guidelines, and processes.

DevOps Lifecycle Integration

GitLab is often referred to as a **complete DevOps platform**, meaning it can support the entire software development lifecycle (SDLC), from planning and development to deployment and monitoring. It enables:

- **Plan:** Use issue tracking, milestones, and boards to plan projects.
- **Create:** Version control for code, and collaborative tools for creating high-quality software.
- **Verify:** Automatically test and validate code with CI pipelines.
- **Package:** Build and package software artifacts for deployment.
- **Deploy:** Automate deployments to production or staging environments.
- **Monitor:** Monitor applications with GitLab's built-in monitoring tools and get feedback to improve code.

GitLab Runners

GitLab Runner is an application that executes CI/CD jobs. When a pipeline is triggered (e.g., after a push to the repository), the runner pulls the repository, executes the defined tasks, and reports back the result. Types of Runners:

- **Shared Runners**
- Available to all projects in the GitLab instance.

Specific Runners

- Dedicated to a particular project or group of projects.

Runners can be installed on different environments (e.g., virtual machines, Docker containers, physical servers), allowing flexibility in how CI/CD jobs are executed.

Security and Compliance

- **Security Testing:** GitLab integrates security tools that automatically scan code for vulnerabilities during the CI pipeline. This includes **SAST** (Static Application Security Testing), **DAST** (Dynamic Application Security Testing), and dependency scanning.
- **Container Scanning:** If your application uses Docker containers, GitLab can scan the container images for vulnerabilities as part of the CI/CD pipeline.
- **Code Review and Approval Processes:** GitLab provides customizable **approval workflows** to ensure that code is reviewed by the right stakeholders before it's merged into production.

Key Features

- **GitLab CI/CD:** Integrated CI/CD pipelines for automated testing and deployment.
- **Version Control:** Git-based version control with repository management.
- **Merge Requests:** Code review and collaboration through merge requests.
- **Issues and Project Management:** Built-in issue tracking and milestone planning.
- **Security:** Security scanning tools integrated into the CI/CD pipeline.
- **DevOps Automation:** Tools to automate every step of the DevOps lifecycle.
- **Container and Kubernetes Integration:** Support for Docker, Kubernetes, and cloud-native deployments.

GitLab is a powerful and versatile platform that integrates every stage of the DevOps lifecycle. Its strong emphasis on **CI/CD**, **project management**, **security**, and **version control** makes it a popular choice for development teams aiming to streamline their workflow, automate processes, and improve collaboration. With options for both cloud-hosted and self-hosted instances, GitLab caters to a wide range of organizations, from small startups to large enterprises.

Continuous Integration / Continuous Deployment (CI/CD)

A **CI/CD pipeline** (Continuous Integration/Continuous Delivery or Continuous Deployment) is an automated process that allows software teams to build, test, and deploy code in a structured, repeatable manner. The pipeline ensures that every change to the codebase is integrated (CI) and delivered or deployed (CD) to production or other environments with minimal manual intervention. This automation improves efficiency, reduces the risk of human errors, and accelerates the delivery of software updates.

CI/CD Pipeline Stages:

Source Control:

- Developers push changes to a version control system (VCS) such as Git.
- A commit or pull request triggers the CI/CD pipeline.

Build:

- The pipeline compiles the source code into an executable format.
- This step may include tasks like dependency resolution, code packaging, and artifact generation.

Testing:

- **Unit Testing:** Small, focused tests are run to ensure that individual units of code function as expected.
- **Integration Testing:** Verifies that different modules or components work together.
- **End-to-End Testing:** Simulates the user experience by testing the complete workflow of the application.
- **Security Testing:** Tools such as Static Application Security Testing (SAST) or Dynamic Application Security Testing (DAST) scan the code for vulnerabilities.
- **Performance Testing:** Ensures the application can handle expected workloads.

Artifact Management:

- If the build passes all tests, the pipeline stores the artifacts (e.g., binaries, Docker images) in an artifact repository (e.g., Nexus, JFrog Artifactory).

Deployment:

- The application is automatically deployed to one or more environments (e.g., staging, testing, production).
- In **continuous delivery**, this may include manual approval before deploying to production. In **continuous deployment**, the process is fully automated.

Monitoring and Feedback:

- The pipeline integrates monitoring tools (e.g., Prometheus, Grafana, New Relic) to ensure the application is performing as expected after deployment.
- Errors or performance degradation trigger alerts or automatic rollbacks if configured.

Deployment of Services

AGILEHAND uses self-hosted Keycloak and GitLab instances deployed for the self-managed authentication and source code storage. Both components are deployed as Kubernetes deployments, Figure 10 demonstrates the deployment, connected services and their interactions with the host system, each other, and external entities.

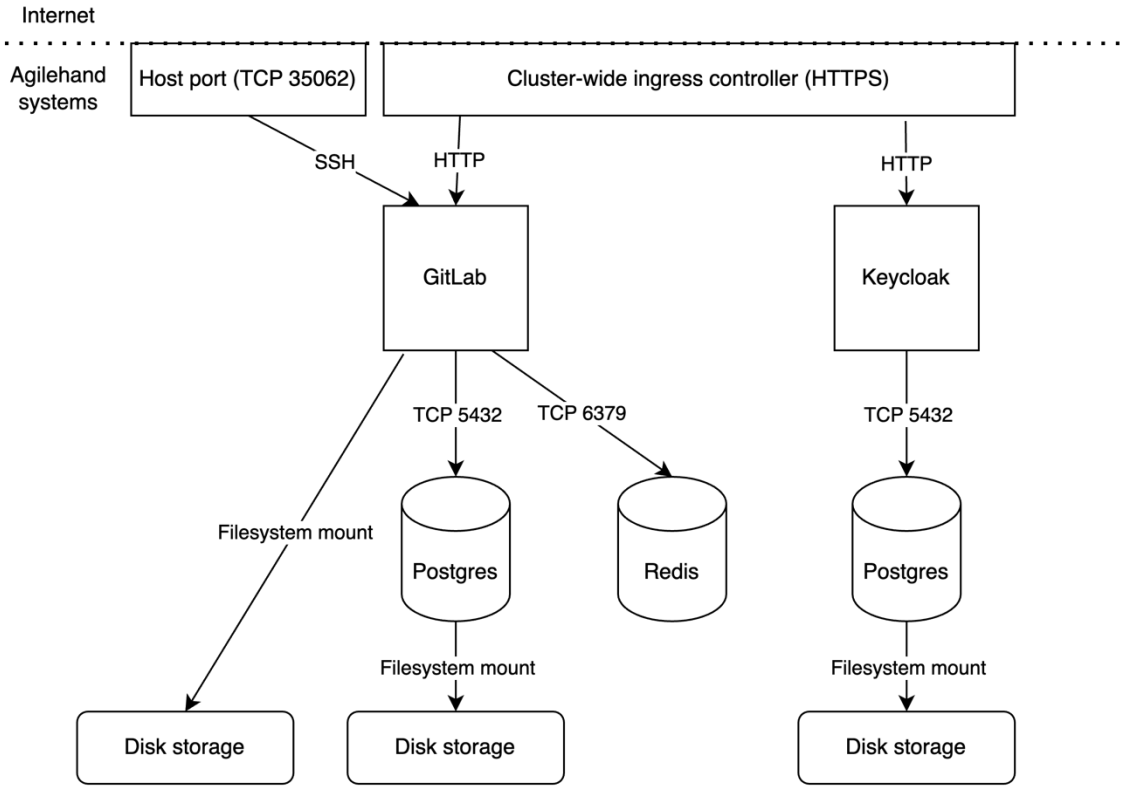

Figure 10 - Deployment of Gitlab and Keycloak services within a kubernetes cluster

Keycloak Deployment

Keycloak is an open-source identity and access management solution. For the AGILEHAND, Keycloak is deployed within a Kubernetes cluster, ensuring secure and scalable authentication and authorization services. This authentication is central point of authentication and entry for all the solutions within the AGILEHAND ecosystem. The setup for Keycloak is available under the AGILEHAND repository.

Components

Keycloak Instance: The core component responsible for handling authentication and authorization requests.

Postgres Database: A relational database used by Keycloak to store user data, configurations, and other relevant information.

Deployment Architecture

Users interact with the Keycloak instance through a cluster-wide ingress controller. The ingress controller manages HTTP requests from external users and forwards them to the Keycloak instance. The Keycloak instance communicates with its Postgres database via TCP. The database is hosted on the same Kubernetes cluster and stores its data on the host system through a filesystem mount. The Postgres database's data is persisted on the host system, ensuring data durability and reliability.

Workflow

External users send authentication requests to the ingress controller. The ingress controller forwards these requests to the Keycloak instance. Keycloak processes the requests and interacts with the Postgres database to validate user credentials. Keycloak sends the authentication response back to the user through the ingress controller.

Gitlab Deployment

GitLab is a comprehensive DevOps platform providing source code management, CI/CD, and more. For the AGILEHAND Project, a self-hosted instance of GitLab is deployed within a Kubernetes cluster to manage the project's codebase and development workflows. Figure 10 demonstrates the GitLab instance and connected services in a Kubernetes deployment. The setup for GitLab is available under the AGILEHAND GitLab repository at agilehand/stack-git.

Components

GitLab Instance: The main application handling source code management, CI/CD pipelines, and other DevOps functionalities.

Postgres Database: A relational database used by GitLab to store project data, user information, and configurations.

Disk Storage: Persistent storage for GitLab's data, such as repositories and artifacts.

SSH Endpoint: An external endpoint for secure shell access to the GitLab instance.

Deployment Architecture

Similar to the Keycloak, GitLab also receives external requests through a cluster-wide ingress controller, which manages HTTP requests. The GitLab instance communicates with its Postgres database via TCP. The database is hosted on the same Kubernetes cluster and stores its data on the host system through a filesystem mount. An additional external endpoint is configured for SSH access, allowing secure interactions with the GitLab instance. GitLab's data, including repositories and CI/CD artifacts, is stored on the host system through a filesystem mount.

Workflow

External users send requests to the ingress controller. The ingress controller forwards these requests to the GitLab instance. GitLab processes the requests, interacting with the Postgres database and Redis instance as needed. Users can also interact with GitLab via SSH for secure operations. GitLab sends the response back to the user through the ingress controller.

Project Template

A template project has been provided which follows a standardized structure for organizing and managing project files, ensuring consistency and ease of navigation. This template includes predefined directories for source code, documentation, configuration files, and other essential components. Team members can replicate the template project to start working on it in a new repository, facilitating efficient collaboration and reducing the likelihood projects following

different folder structures. The template also incorporates best practices for file naming conventions and directory organization, promoting a clean and professional project layout that supports scalable and maintainable development. The project template is available in the root directory of AGILEHAND group in AGILEHAND GitLab repository.

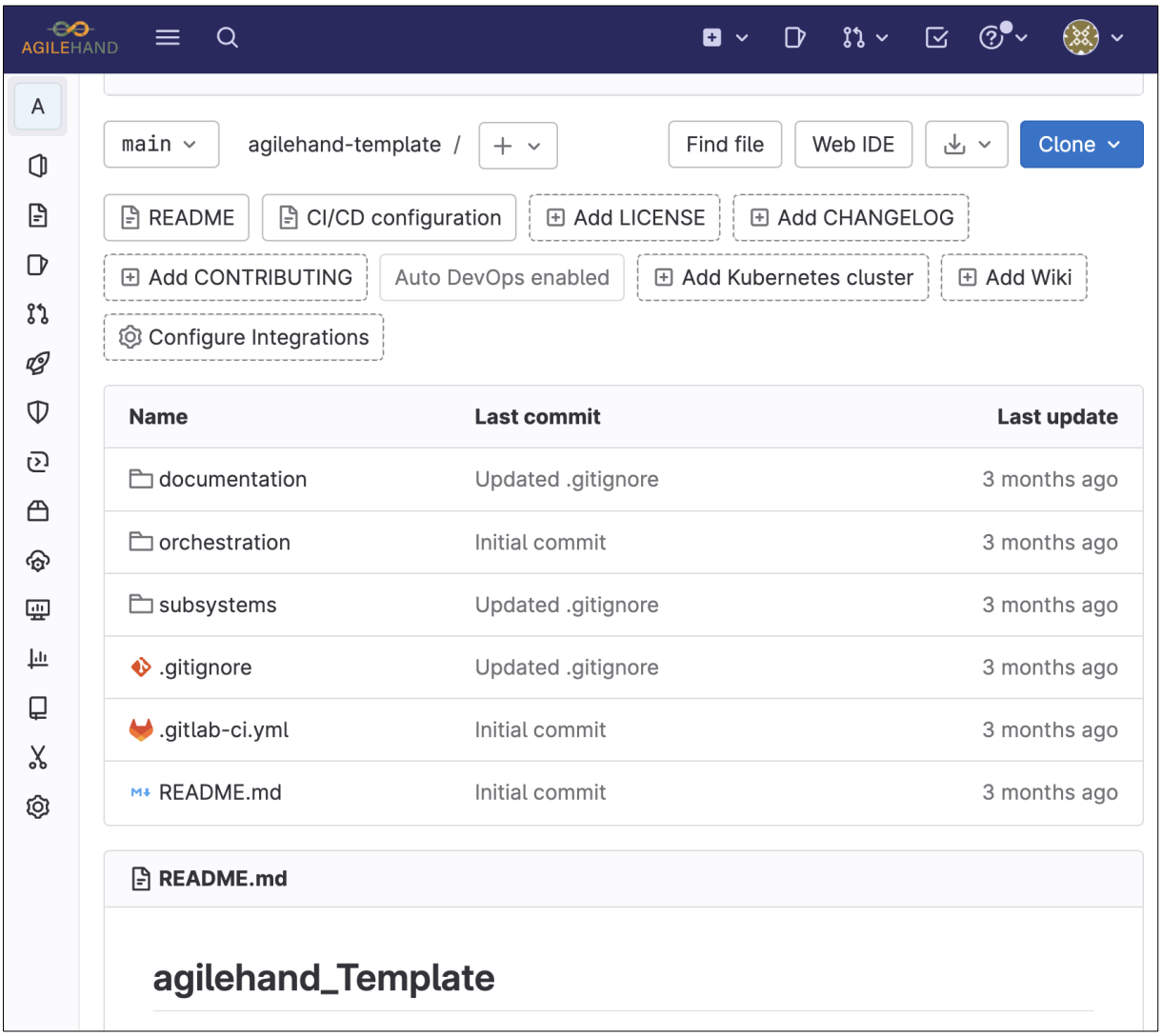

Figure 11 - AGILEHAND template project in Gitlab

Contribution Guidelines

The contribution guidelines for the AGILEHAND Project outline the process and standards for contributing to the project's codebase. These guidelines include instructions for setting up the development environment, coding standards, and the procedure for submitting pull requests. Contributors are encouraged to follow these guidelines to ensure that their contributions are consistent with the project's quality and style requirements. The guidelines also emphasize the importance of writing clear commit messages, conducting thorough code reviews, and adhering to the project's code of conduct. By following these guidelines, contributors can help maintain the integrity and quality of the project while fostering a collaborative and respectful development environment. These guidelines can be found at the root directory of the AGILEHAND Gitlab Repository.

4. Conclusion

The deliverable D3.3 provides a detailed overview of the design, deployment, and integration of various solutions within the AGILEHAND project. It highlights the architectural framework, technical specifications, and integration strategies essential for creating a cohesive and scalable system tailored to agile and reconfigurable manufacturing environments. The platform bridges the gap between standalone and cloud-based solutions, offering centralized management while maintaining the autonomy of each suite. Key components like Single Sign-On (SSO), role-based access control, and a modular framework contribute to a user-friendly experience that meets industry standards for security and flexibility. Use of technologies like Keycloak for authentication, GitLab for source code management, and Superset for data visualization underscores the project's commitment to leveraging open-source technologies. These tools, integrated within a Kubernetes-based infrastructure, ensure scalability, reliability, and ease of management. Pilot use case has been demonstrated for Produmar Pilot, highlighting the practical applications and transformative potential of AGILEHAND solutions in real-world contexts, enhancing efficiency, reducing costs, and improving product quality. The iterative development and validation of these solutions across pilot scenarios highlight the project's focus on aligning technical architecture with stakeholder needs, ensuring modularity, interoperability, and user-centric design. Overall, D3.3 showcases the strategic vision and technical capabilities of the AGILEHAND project, emphasizing its potential to revolutionize manufacturing processes through innovative, integrated, and scalable solutions.