MARI4YARD

MARI4ALLIANCE

INESCTEC

iilab INDUSTRY & INNOVATION LAB

Internal training:

Mobile Manipulation for Internal Logistics

November 22 and 23, 2023

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

iiLab's Mission and Objectives

- ➢ Demonstration of concepts and advanced technologies in the areas of robotics, automation, industrial cyber-physical systems (Internet of things).
- ➢ Dissemination of INESC TEC's expertise for the industry and the community in general.
- ➢ Experimentation and prototyping space for technological companies
- ➢ Tailor-made training for senior managers and senior executives of industrial companies

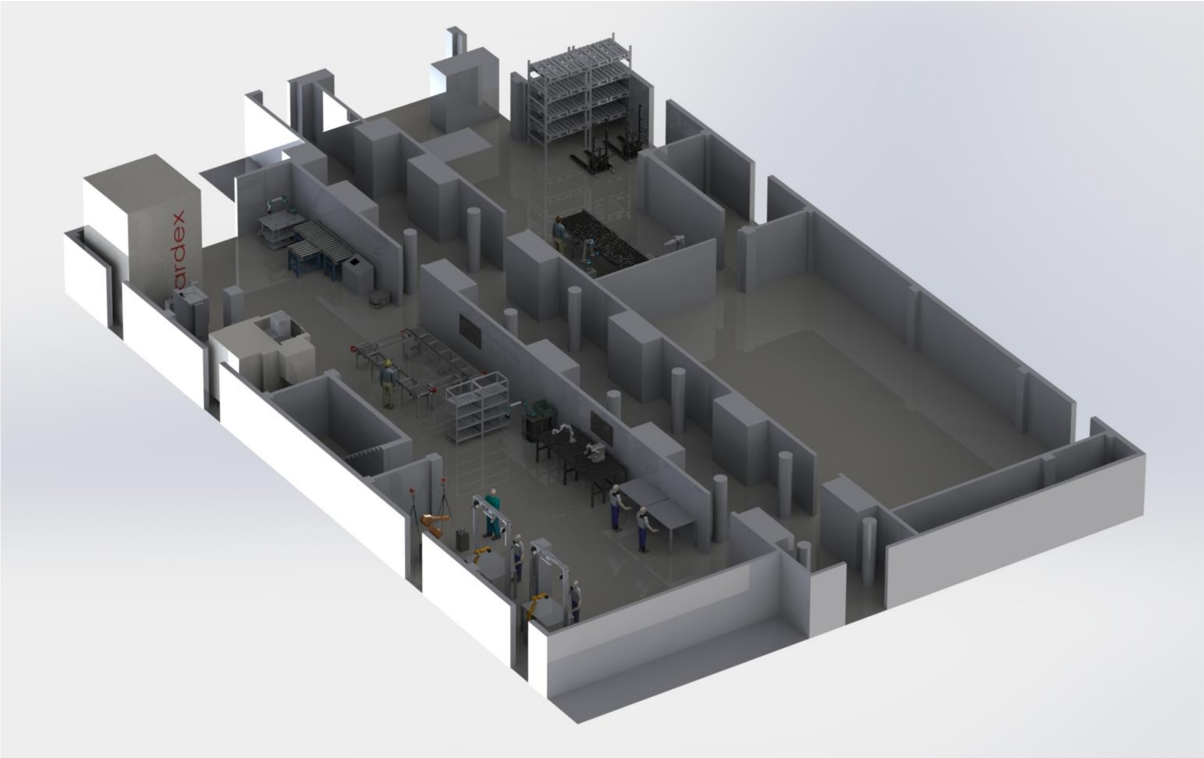
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

iilab INDUSTRY & INNOVATION LAB

MARI4YARD
MARI4ALLIANCE

- Open Space (up to 30 researchers)
- Training Room
- Industrial Premises (350 m2)
- Mechanical/Electrical Workbench



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Mobile Manipulator for Internal Logistics

Luis Rocha
Senior Researcher
INESC TEC

November 22 and 23, 2023

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC iilab INDUSTRY & INNOVATION LAB

MARI4YARD MARI4ALLIANCE

Motivation

➢ At today's shipyards, the transportation of raw materials and/or manufactured parts between stores and workshops, and from workshops to subassembly areas, is still heavily reliant on human operators.

➢ This transportation is typically performed by hand or by using self-propelled, pulled, or pushed platforms.

➢ During the shipbuilding process a wide range of components including structural steel, pipes, cables, valves, and outfitting are supplied, handled, and transported. These parts are normally stored in warehouses or pallets, and are placed in shelves, big containers and/or boxes.

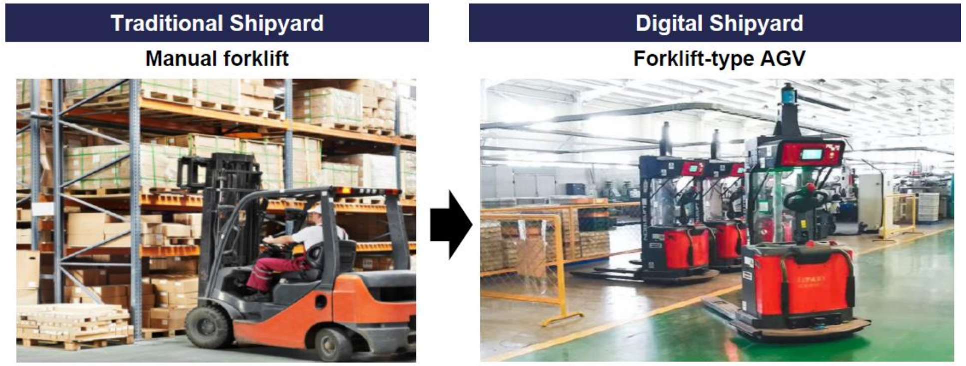
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Motivation

➢ From the state-of-art in mobile robotics, it is possible to find several commercial AGVs/AMRs solutions that could be used by shipbuilders to automate some of their logistic tasks*.

➢ These solutions, though, present limitations regarding the manipulation of the loads.

- ➢ automate forklifts are able to directly pick pallets for transportation
- ➢ more general AGV/AMR solutions required the addiction of a transfer system to enable the load to be automatically transferred from the place that it is stored.
- ➢ However, they are not able to select and pick individual parts form containers or bins.



*John Spoehr, Ryan Jang, Kosta Manning, Arvind Rajagopalan, Cecilia Moretti, Ann-Louise Hordacre, Sara Howard, Peter Yaron and Lance Worrall **The Digital Shipyard, Opportunities and Challenges, March 2021, Flinders University - Australian Industrial Transformation Institute**

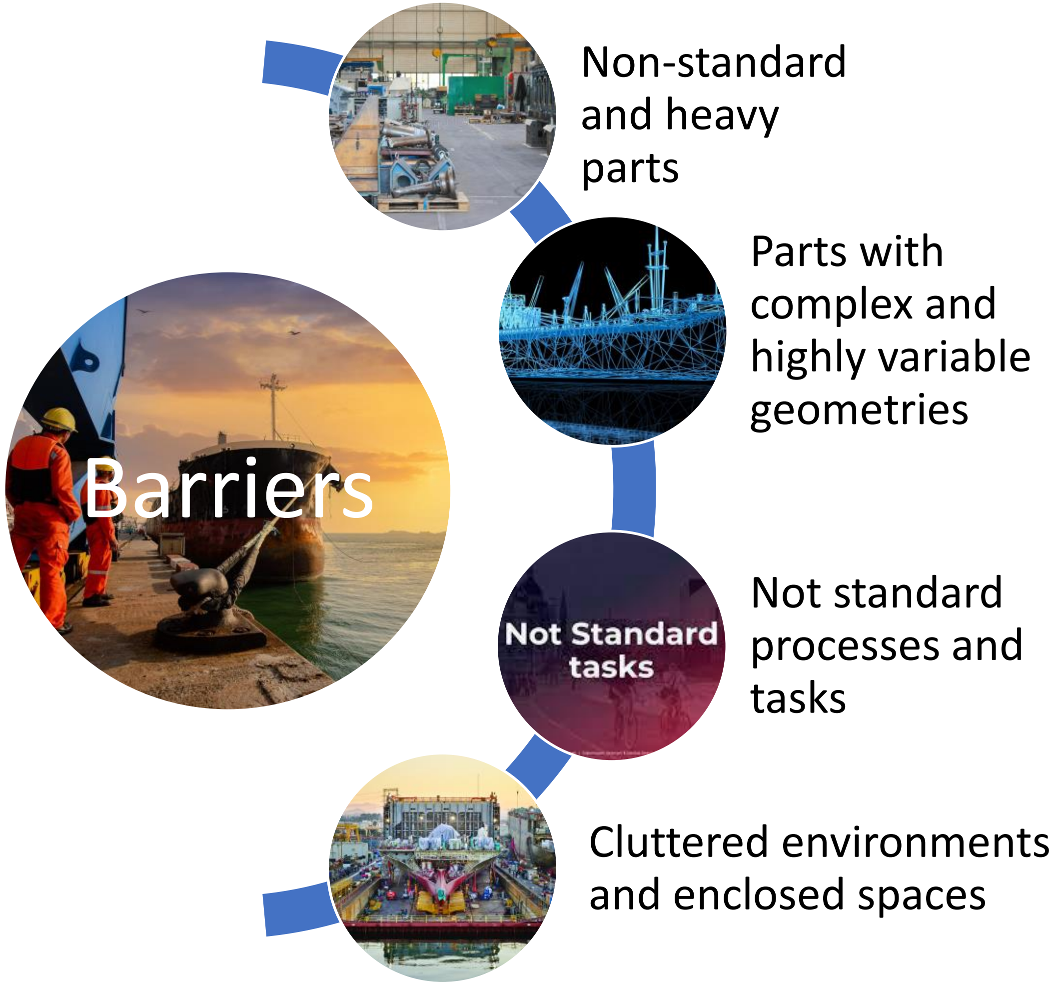
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC
iilab INDUSTRY & INNOVATION LAB

MARI4YARD
MARI4ALLIANCE

Challenges

➢ The introduction of mobile robotics into shipbuilding processes is hindered by several factors, including:



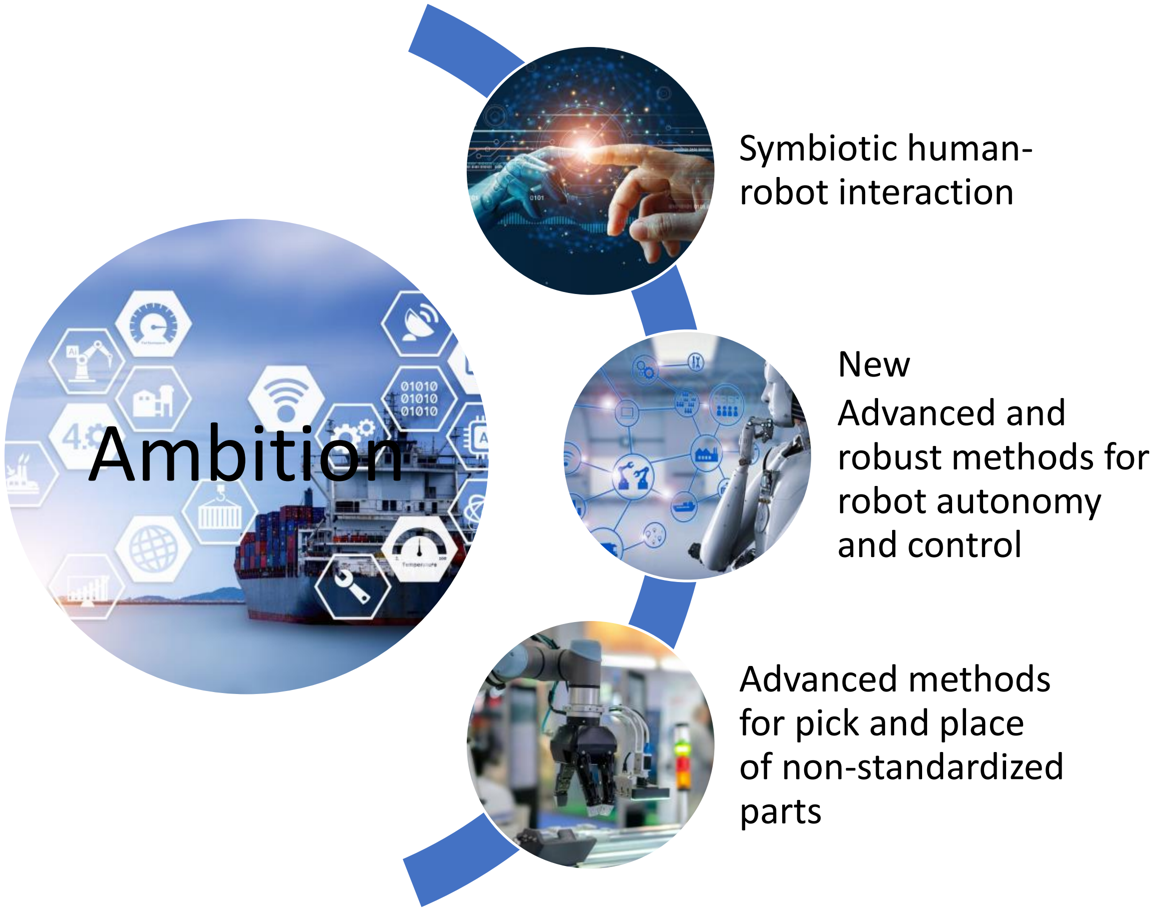
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC iilab INDUSTRY & INNOVATION LAB

MARI4YARD MARI4ALLIANCE

Ambition

➢ The effective implementation of autonomous mobile robots needs to be tailored to the specific demands of the shipbuilding industry and its processes, requiring further developments.

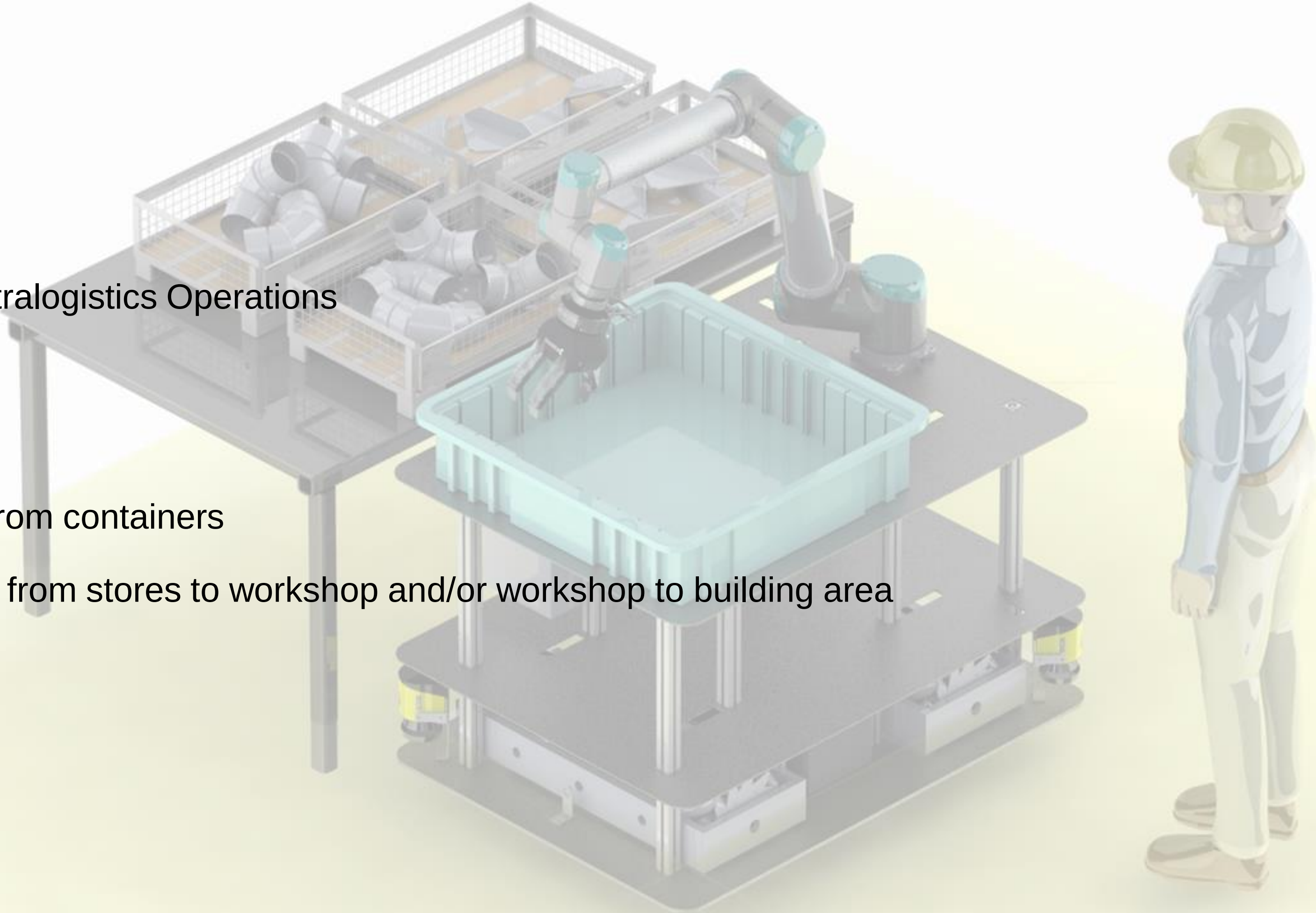


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Value Proposition

- Mobile Manipulator for Intralogistics Operations

- Able to autonomously:
  - Pick individual parts from containers
  - Transport them parts from stores to workshop and/or workshop to building area

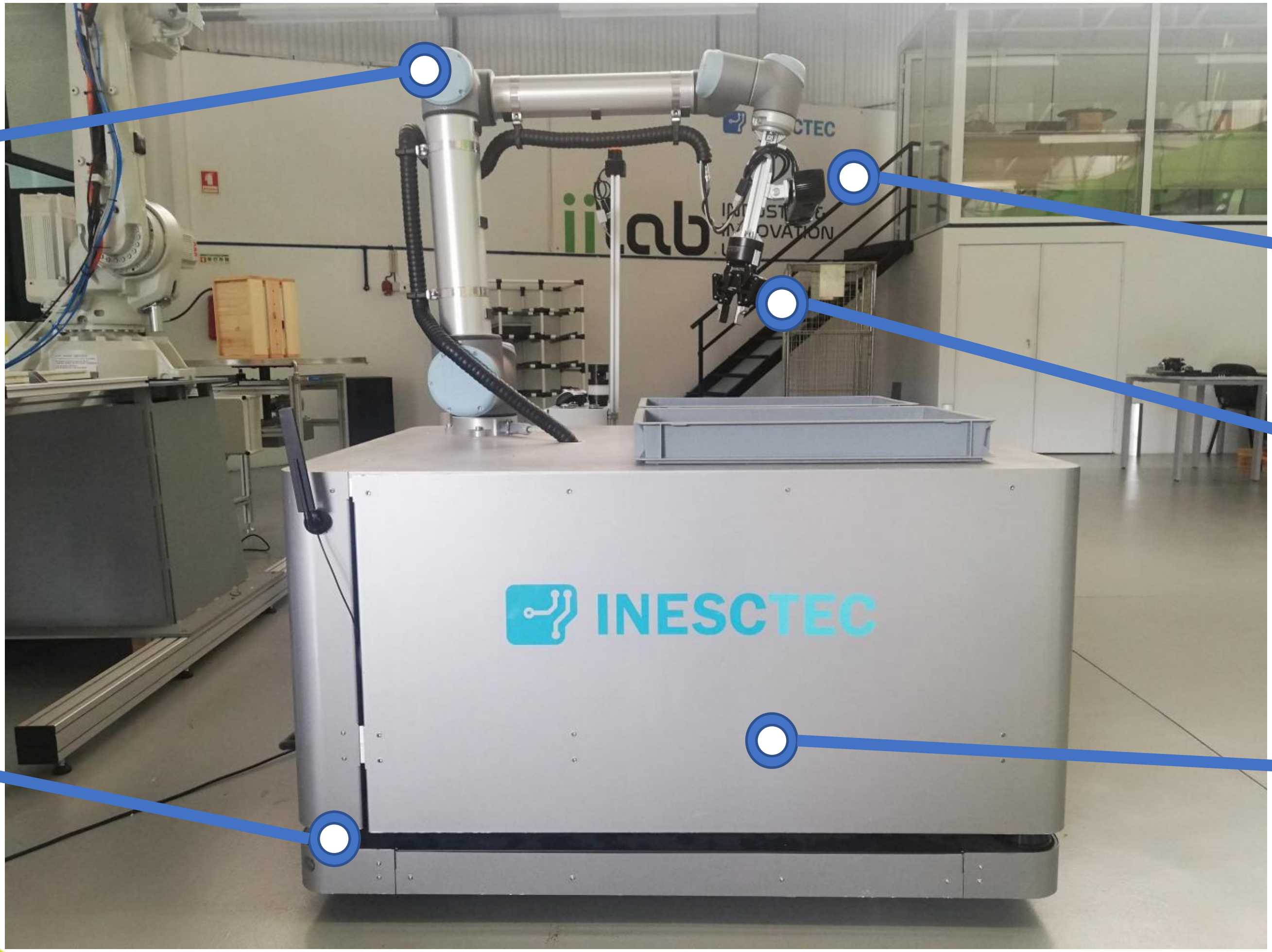


This project has received funding from
the European Union’s Horizon 2020
research and innovation programme
under grant agreement No 101006798

Mobile Robotic Platform

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Mobile Robotic Platform



Cobot

RGB-D Sensor

2 Finger Gripper

Safety Laser Scanner

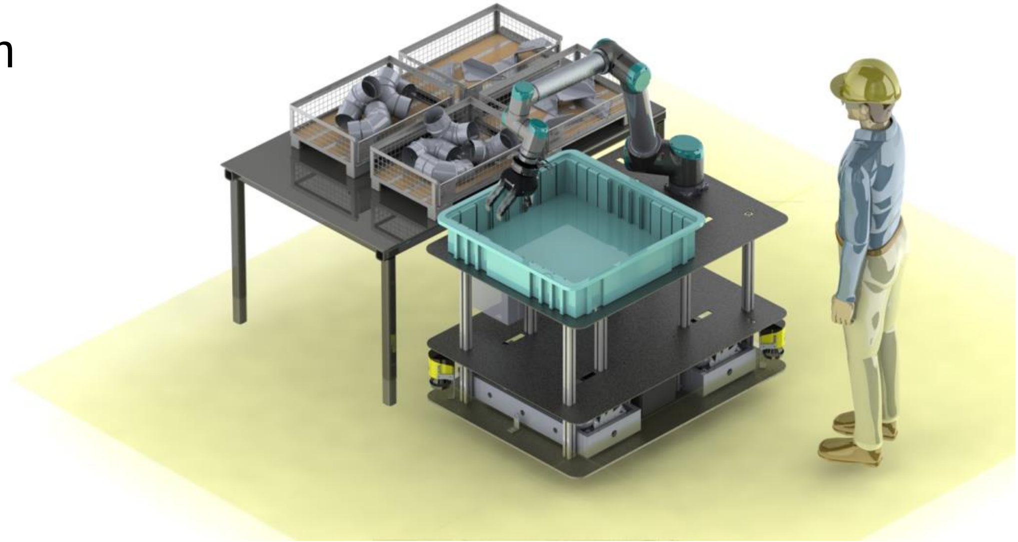
Mobile Platform Omnidirectional

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

MARI4YARD
MARI4ALLIANCE

Technical Overview

MRO	**Mobile** robotic platform + **Collaborative** robot arm
PPM	**Process Perception**
WMS	**Workspace Monitoring**
COP	**Control Orchestration and Planning**
HRIM	**HRC** – Human Robot Interaction Mechanisms



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Agenda / Objectives
Day 1

3.1 Module 1

TITLE	Mobile Robot Navigation System Configuration
LANGUAGE	English
DESCRIPTION	This module aims to present the entire process of installing and configuring a mobile robot from scratch. From mapping a new space using SLAM algorithms to defining trajectories.
PRE-REQUISITES	Basic knowledge of informatics.
LEARNING OUTCOMES	After completing this module, the learner will be able to: • To understand the fundamentals of a mobile robot's operation. • To learn the configuration and installation of a mobile robot through demonstration. • To understand the concept of mapping using SLAM algorithms. • To understand the concept of trajectory definition in mobile robot applications.
LEARNING CONTENT	Mobile robot navigation, simultaneous localization and mapping (SLAM algorithms), mobile robot localization and controllers, trajectory definition/generation.
APPROACH/ METHOD	The introduction to this topic will be based on the observation and experimentation of configuring a real robot's navigation system at the iiLab facilities.
DELIVERY FORMAT	In-person
DURATION	2 x 1h20
INFRASTRUCTURE	• Learning factory • Mobile manipulator
DATE	22 November 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Agenda / Objectives
Day 1

3.2 Module 2

TITLE	OSPS - Open Scalable Production System
LANGUAGE	English
DESCRIPTION	This module aims to acquaint learners with the OSPS stack and teach them to use it for skill-based robot programming.
PRE-REQUISITES	The learner should have: • Basic Knowledge of ROS • Experience programming in Python (preferable) or C++
LEARNING OUTCOMES	After completing this module, the learner will be able to: • Understand the advantages of skill-based robot programming. • Recognize the main components of the OSPS stack. • Use the Skill Generator and create custom robotic skills. • Configure a Task Manager instance and understand its message-based internal API. • Use the Production Manager to monitor Task Manager instances. • Assemble robotic tasks using the Task Creator. • Execution of robotic tasks using the Production Manager and control and monitor their execution.
LEARNING CONTENT	• Overview of the OSPS stack. • OSPS messages. • Skill Generator. • Task Manager. • Production Manager. • Task Creator.
APPROACH/ METHOD	Each lecture concept is accompanied by a theoretical explanation and a practical session conducted in parallel.
DELIVERY FORMAT	In-person
DURATION	2 x 1h20
INFRASTRUCTURE	• Training room. • Notebooks.
DATE	22 November 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Agenda / Objectives
Day 2

3.3 Module 3

TITLE	Robotic Grasping 101
LANGUAGE	English
DESCRIPTION	Object grasping and manipulation are essential tasks in robotics, allowing robots to interact with their environment and perform various operations, from simple pick-and-place operations to complex assembly and manufacturing processes. By automating these tasks, robots can increase efficiency, reduce costs, and improve safety in the workplace. In the context of the Mari4_Yard project, INESC TEC developed a robotic object-grasping solution aiming to grasp operations in shipyards by using a mobile manipulator. This module will present a theoretical background in automated grasping tasks, giving a general overview of sensing and handling object techniques from hardware to software perspectives.
PRE-REQUISITES	No prior knowledge or competence is required.
LEARNING OUTCOMES	After completing this module, the learner will be able to: • List different gripper technologies. • Categorize grasping techniques. • Differentiate grasping approaches. • List different sensing technologies. • Describe object recognition strategies. • Define a grasping mission.
LEARNING CONTENT	• End-effectors definition and types. • Grasping modelling. • Sensing technologies. • Object recognition strategies.
APPROACH/ METHOD	This module will be based on a fully expository lecture. Theoretical concepts will be presented with basic robotic grasping system concepts to be later deployed in Module 6.
DELIVERY FORMAT	In-person
DURATION	1h20
INFRASTRUCTURE	• Training room
DATE	23 November 2023

3.4 Module 4

TITLE	Robotic Grasping Hands-on
LANGUAGE	English
DESCRIPTION	The current module presents to the participants a hands-on section allowing the deployment of a robotic grasping application based on the developed system in the context of the Mari4_Yard project.
PRE-REQUISITES	• Modules 4 and 5.
LEARNING OUTCOMES	After completing this module, the learner will be able to: • Build a grasping mission using a mobile manipulator. • Evaluate the application limitation and applicability.
LEARNING CONTENT	• Design the overall mission. • Build the dataset. • Tests on a mobile robot.
APPROACH/ METHOD	The participants will be invited to a hands-on session deploying concepts presented in modules 4 and 5.
DELIVERY FORMAT	In-person
DURATION	1h20
INFRASTRUCTURE	• Learning Factory. • Omnidirectional mobile manipulator. • Notebooks.
DATE	23 November 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Agenda / Objectives
Day 2

MARI4YARD
MARI4ALLIANCE

3.5 Module 5

TITLE	AR-based Human-Robot Interaction
LANGUAGE	English
DESCRIPTION	An essential factor of the current manufacturing paradigm is the emphasis on the human aspect, characterized by the interaction between humans and machines. In this context, augmented reality can assist the operator to cope with the flexibility and adaptability inherent to the Industry 4.0. This training session will cover the basics of extended reality and human-robot interaction, including theoretical background and use of an AR application in a head-mounted display. Throughout the module, participants will work on hands-on exercises that will help them gain practical experience in programming, deploying, and running a complete task on a collaborative robot without coding. They will also have the knowledge and skills to continue exploring and building upon what they have learned as they program new robot programs.
PRE-REQUISITES	No prior knowledge or competence is required.
LEARNING OUTCOMES	After completing this module, the learner will be able to: • Describe the main human-robot interaction paradigms. • Explain the benefits of human-robot interaction. • Describe the main concepts related to extended reality. • Deploy an application on a head-mounted display. • Use a head-mounted display to interact with a collaborative robot.
LEARNING CONTENT	• Extended reality. • Head-mounted displays. • Human-robot interaction.
APPROACH/ METHOD	This module will start with a short theoretical session where the main concepts will be presented. Next, participants will be invited to the Learning Factory for a hands-on session, in which they will use a head-mounted display to program a collaborative robot.
DELIVERY FORMAT	In-person
DURATION	1h20
INFRASTRUCTURE	• Training room. • Learning factory. • Omnidirectional mobile manipulator. • Head-mounted display.
DATE	23 November 2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Thank you for your attention!

Luis Rocha

luis.f.rocha@inesctec.pt

This project has received funding from
the European Union’s Horizon 2020
research and innovation programme
under grant agreement No 101006798

Internal Training

Mobile Manipulation for Internal Logistics

June 11 and 12, 2024

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Module 1

Mobile Robot Navigation System Configuration

June 11 and 12, 2024

Paulo Rebelo
Researcher
INESC TEC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INDEX

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

MARI4YARD
MARI4ALLIANCE

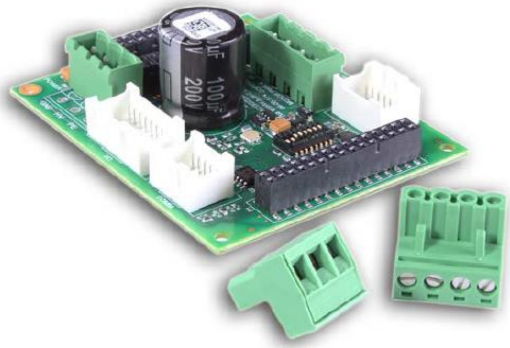
FRIDAY

AUTONOMOUS MOBILE MANIPULATOR

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Hardware



CANopen

INESCTEC

This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

ROBOT HARDWARE - INDEX

A. Security System

1. Lasers
2. PLC
3. Wago Modules

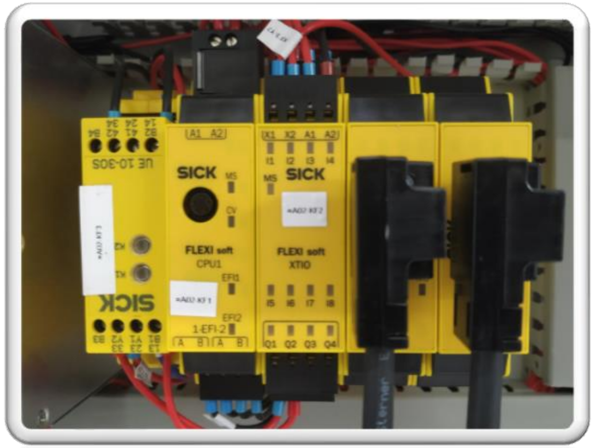
B. Traction System

1. Motors
2. Drivers
3. Encoders

C. Power System

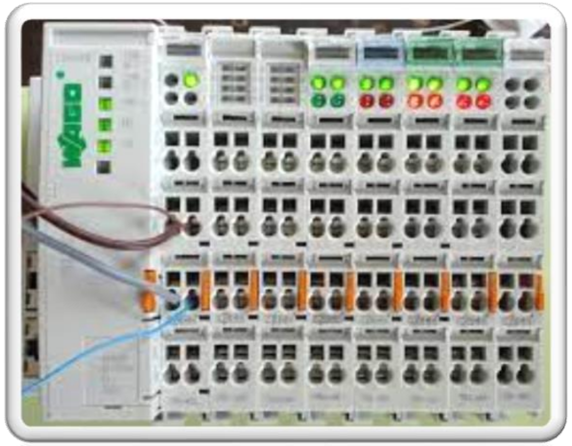
1. Batteries
2. Chargers
3. Electrical Board

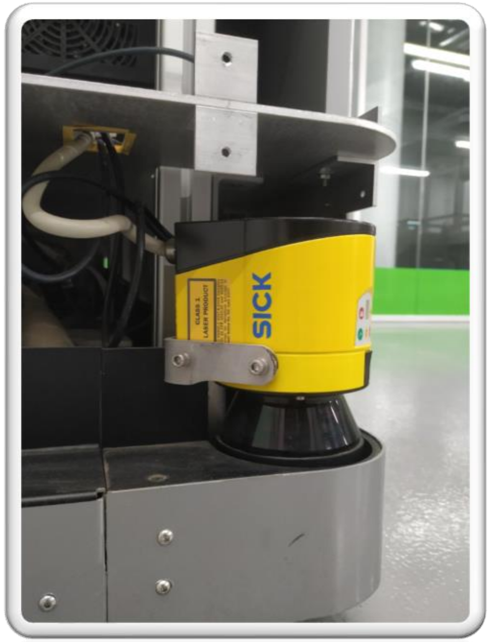
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798



Robot Hardware

Security System





INESCTEC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Security System

- Sick Security Lasers – LiDAR (Light Detection and Ranging) Sensors

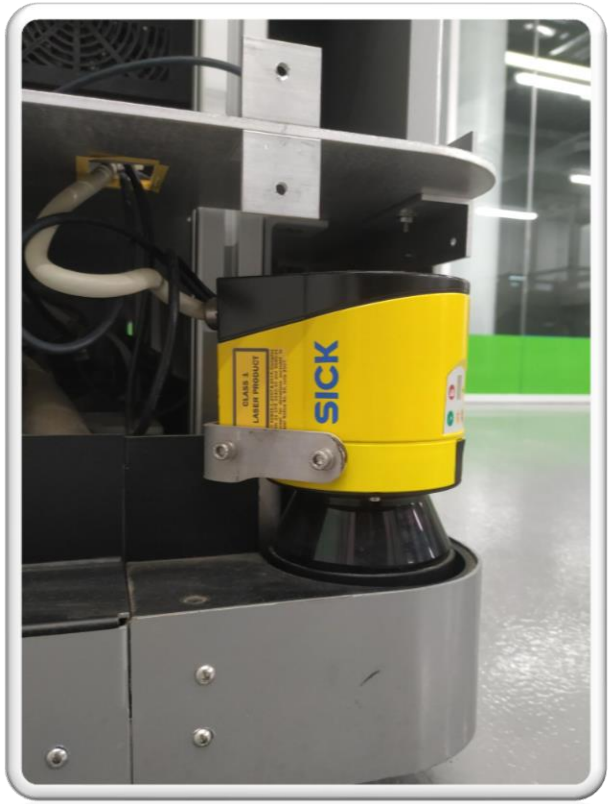
- Provide crucial data for navigation, safety, and obstacle avoidance.
- Ability to create accurate 2D/3D maps of the environment and detect objects in real-time enables mobile robots to operate safely.

Used for:

- Collision Avoidance
- Navigation and Mapping
- Obstacle Detection

Main Characteristics:

- Improved Accuracy and Precision
- Versatility
- Adaptability to Various Lighting Conditions
- Remote Monitoring and Control



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Security System

- PLC

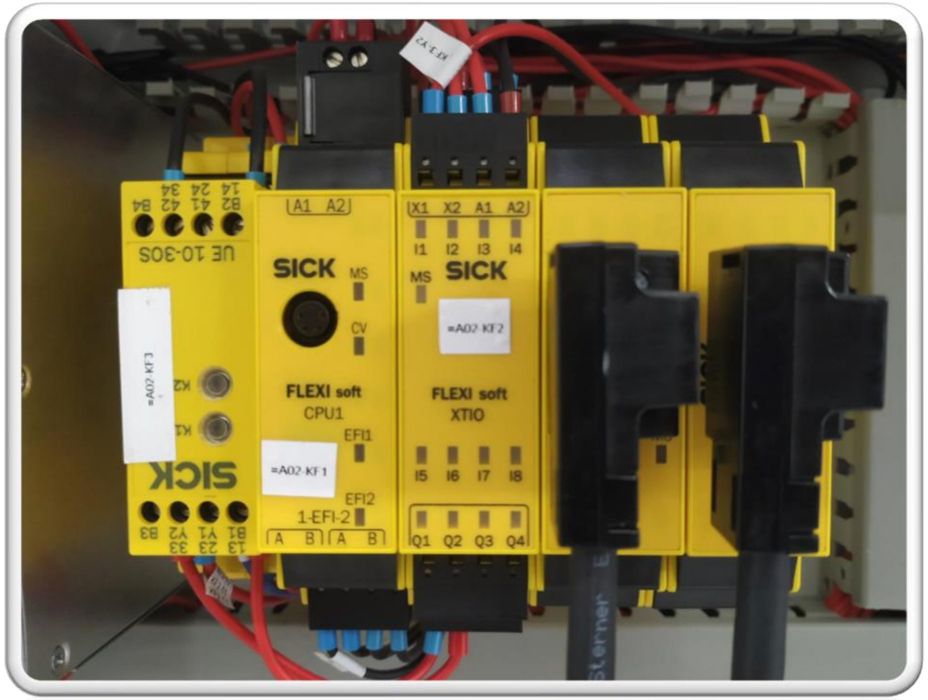
Used for:

- Contribute to the security of mobile robotic systems by managing data from various sensors, implementing safety measures, and integrating with broader security.
- Combined with advanced sensors, like LiDAR, can help ensure the safe and secure operation of mobile robots in various applications.

- Control and Coordination
- Emergency Shutdown and Safety
- Data Processing and Decision-Making

Main Characteristics:

- Integration with Security Systems
- Logging and Reporting
- Remote Monitoring and Control



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Security System

- WAGO Modules

MARI4YARD
MARI4ALLIANCE

- WAGO modules are popular for their reliability and versatility, making them a valuable choice for integrating and controlling various aspects of mobile robotic systems;
- Switch between different **Security Zones** with different characteristics.

Used for:

- Control and Automation
- Modularity and Scalability
- Safety Features

Main Characteristics:

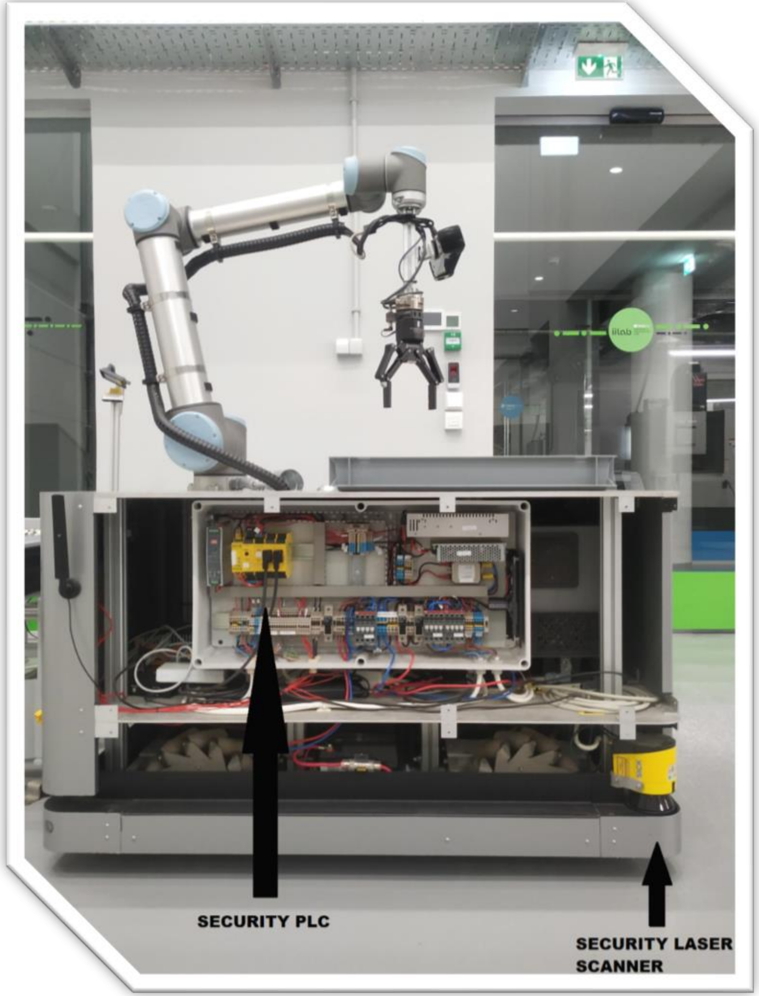
- Sensors and Actuators
- Real-Time Processing
- Safety Features
- Remote Monitoring and Control

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

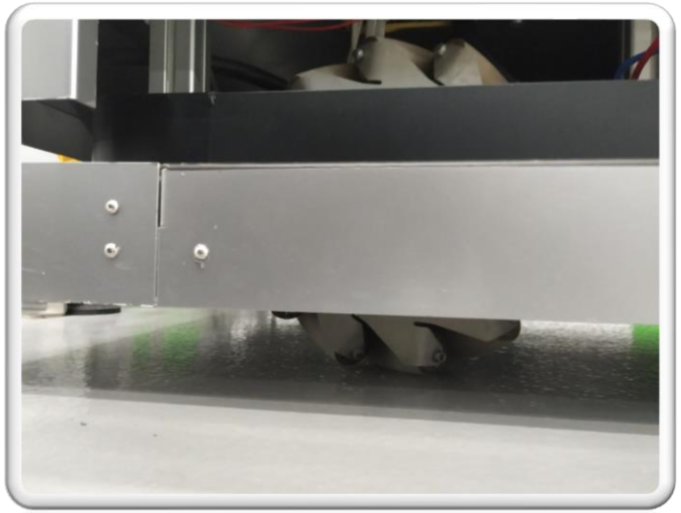
- Security System

- RESUME



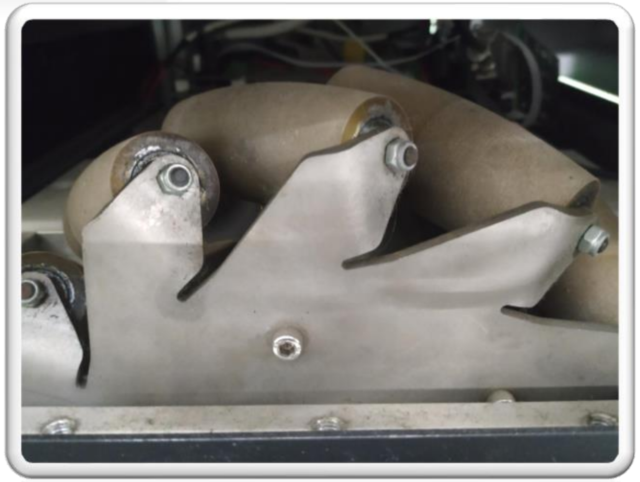
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

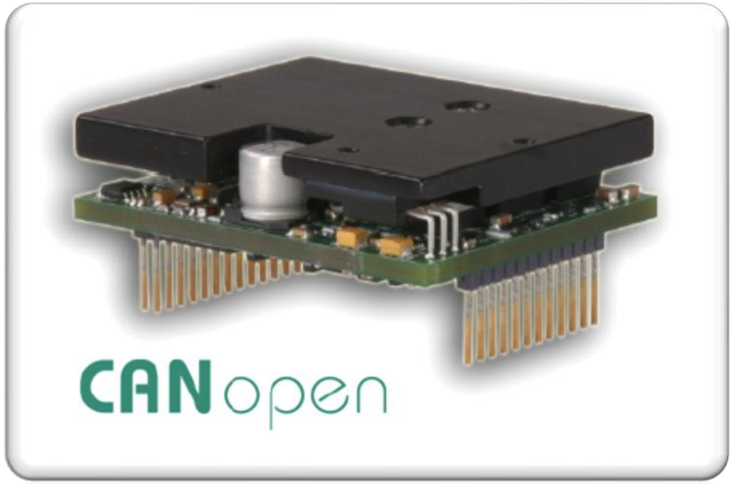
MARI4YARD
MARI4ALLIANCE



Robot Hardware

Traction System





INESCTEC

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Traction System

• AMC (Advanced Motor Controllers) Drivers

MARI4YARD
MARI4ALLIANCE

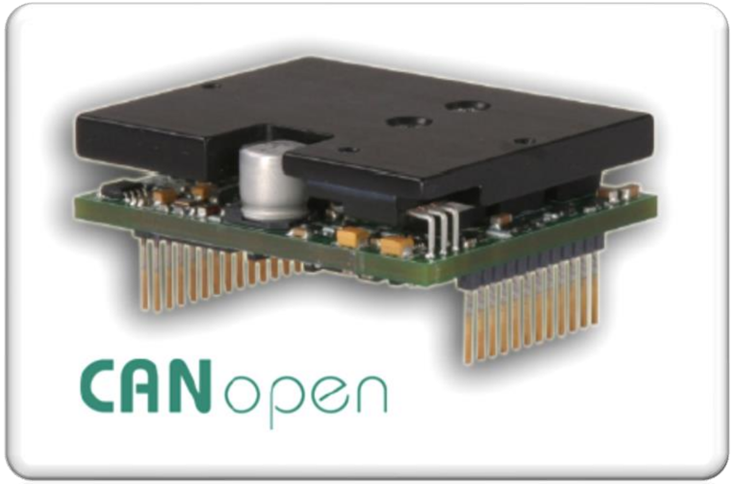
- Crucial for managing the movement of robots, including controlling the speed, direction, and position of motors;
- Ensure that robots perform their intended tasks effectively and efficiently;

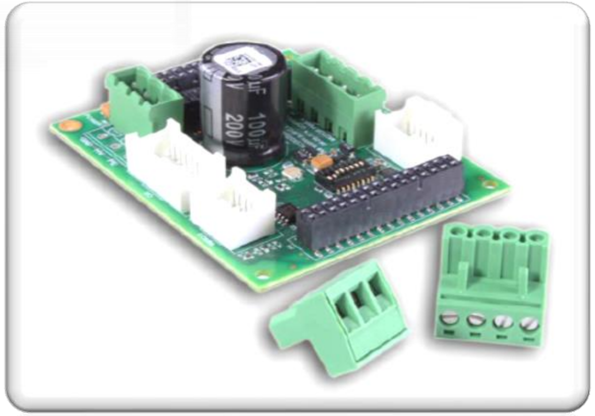
Used for:

- Motor Control
- Sensors Integration
- Customization

Main Characteristics:

- Communication
- Energy Efficiency
- Safety





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Traction System

• Motors

MARI4YARD
MARI4ALLIANCE

- Playing a crucial role in enabling the movement and functionality of robotic systems;
- Provide the mechanical motion necessary for mobile robots to navigate, interact with their environment, and perform various tasks;

Motors Types:

- DC Motors (Brushed or Brushless);
- Stepper Motors;
- Servo Motors;

Main Characteristics:

- Payload Requirements;
- Speed and Efficiency;
- Environment;
- Cost;

- **Brushless DC Motors (BLDC):** BLDC motors have a more durable and efficient design. They are commonly used in mobile robotics due to their higher efficiency, longer lifespan, and lower maintenance requirements. BLDC motors are often found in applications where precision and reliability are essential.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Traction System

• Encoders

MARI4YARD
MARI4ALLIANCE

- Provide essential position, speed, and direction feedback for precise control and navigation;
- Enable mobile robots to move accurately, manipulate objects, and perform tasks with reliability and efficiency

Encoders Types:

- Rotary / Linear;
- Incremental / Absolute;

Main Characteristics:

- Payload Requirements;
- Speed and Efficiency;
- Environment;
- Cost;

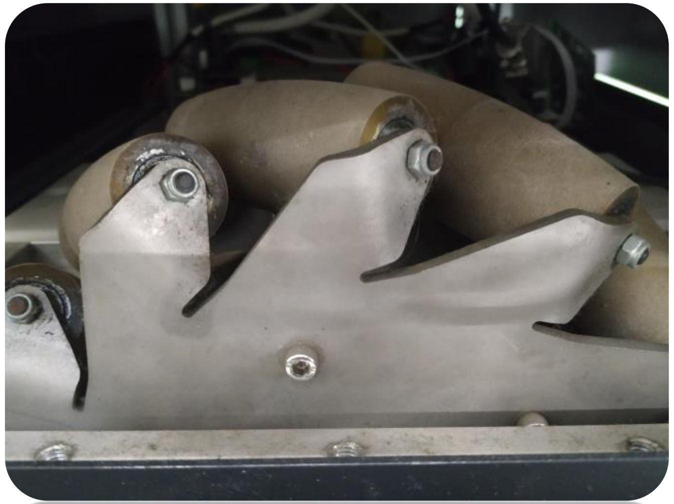
- **Incremental Encoders:** generate a series of pulses (quadrature signals) as the shaft or object rotates, allowing the controller to track movement and calculate position and speed. However, they do not provide absolute position information, so they require a reference point (usually a home position) to establish the initial position.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

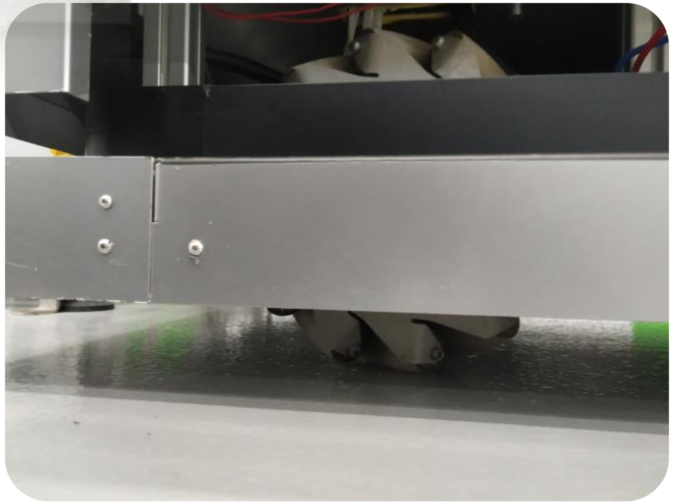
- Traction System

- Mecanum Wheels



MARI4YARD
MARI4ALLIANCE

- Offer an unique and versatile method for achieving both translational and rotational motion without the need for complex steering mechanisms, allowing an omnidirectional movement in mobile robotics;
- Distinctive design allows for precise and flexible control, making them valuable in applications where agility and maneuverability are crucial.



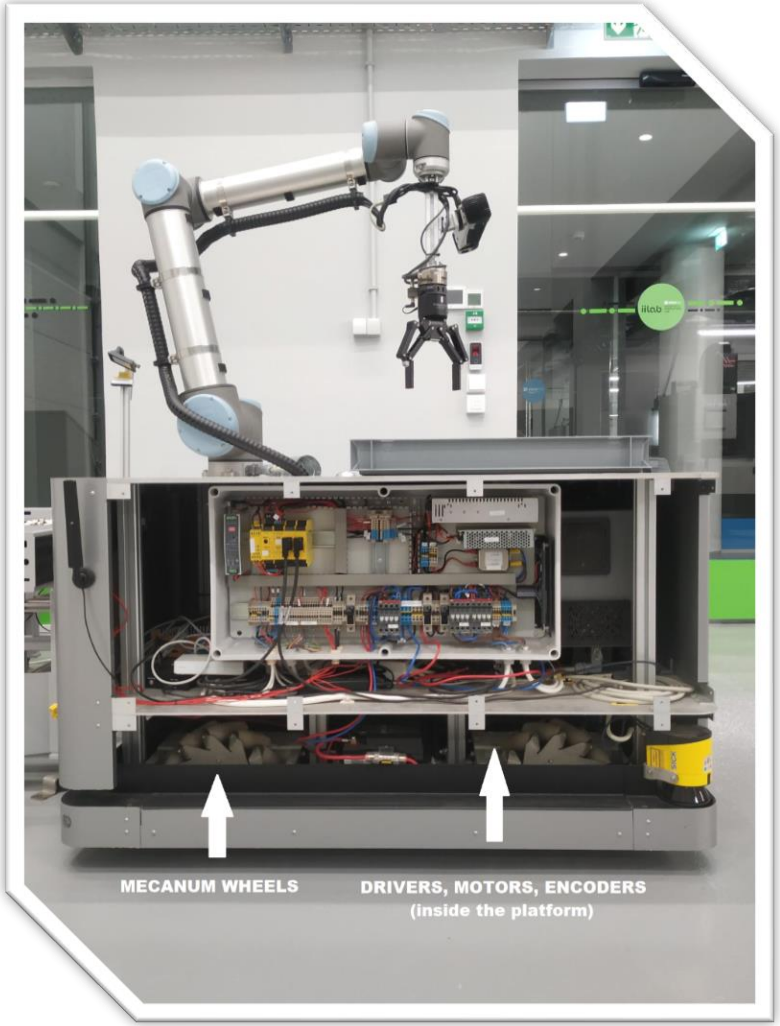
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

- Traction System

- RESUME

MARI4YARD
MARI4ALLIANCE

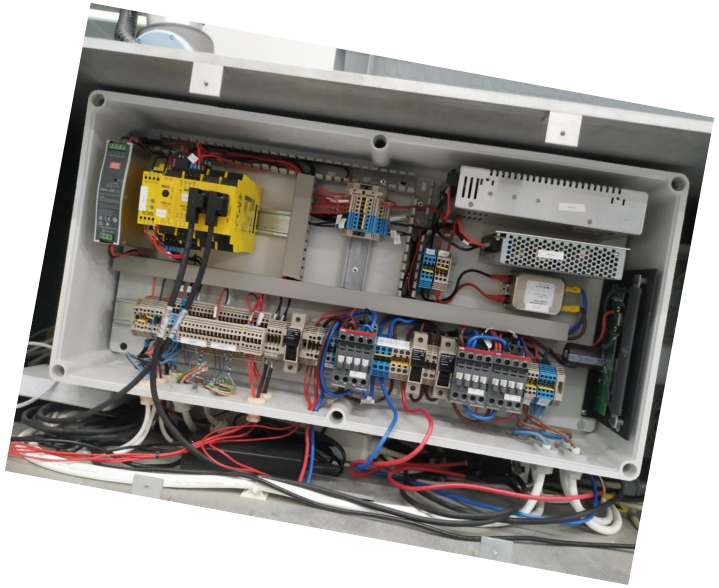


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Hardware

Power System



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Power System

- Batteries

MARI4YARD
MARI4ALLIANCE

- Considered a critical power source for mobile robotics, influencing the robot's mobility, endurance, and overall performance
- Provide the mechanical motion necessary for mobile robots to navigate, interact with their environment, and perform various tasks;

Battery Types:

- Lithium-Ion (Li-ion)
- Lead-Acid
- Nickel-Based

Main Characteristics:

- Energy Density;
- Safety;
- Durability;
- Management;

- **Li-ion Batteries** are widely used in mobile robotics due to their high energy density, lightweight nature, and relatively low self-discharge rates. They have a good balance between energy storage capacity and weight.
- **Li-ion Batteries** are often preferred in applications where mobility and endurance are critical.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

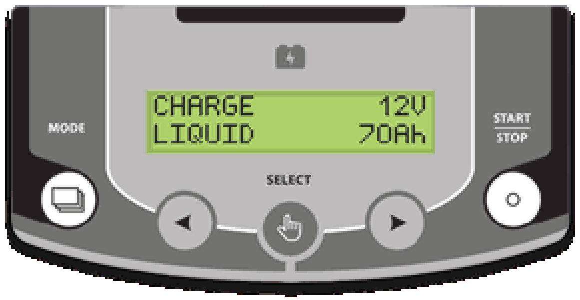
INESCTEC

• Power System

• Chargers

MARI4YARD
MARI4ALLIANCE

- **Chargers** play a crucial role in the power management and operation of mobile robotic systems by replenishing the energy stored in batteries. The design and capabilities of chargers significantly impact the efficiency, reliability, and overall performance of mobile robots;
- Advancements in charging technologies, including **fast charging** and **smart charging** solutions, continue to contribute to the development of more capable and autonomous mobile robotic systems.
- Nowadays the most commonly used chargers are **wireless chargers**.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

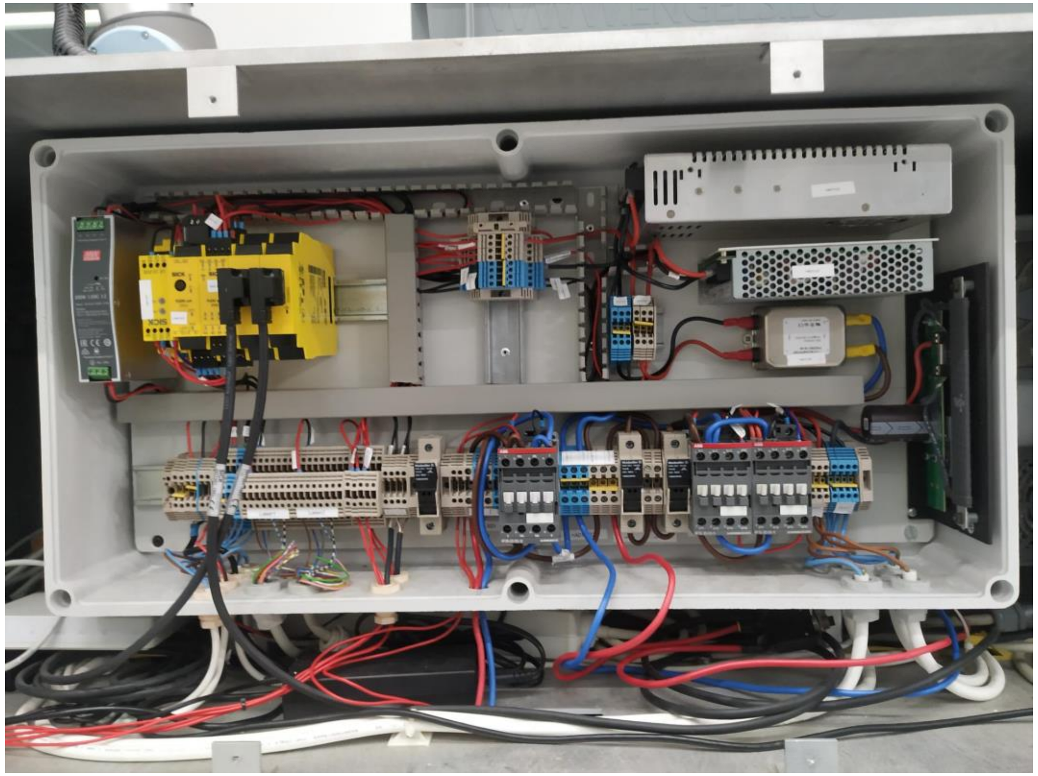
INESCTEC

• Power System

- Electrical Board

MARI4YARD
MARI4ALLIANCE

- A **Power Distribution Board** (PDB) is a **type of electrical board** designed to distribute power from a main power source (e.g., battery) to various components within a system. It typically includes connections for power input, output terminals for devices, voltage regulation, and sometimes features like current monitoring or protection circuits.



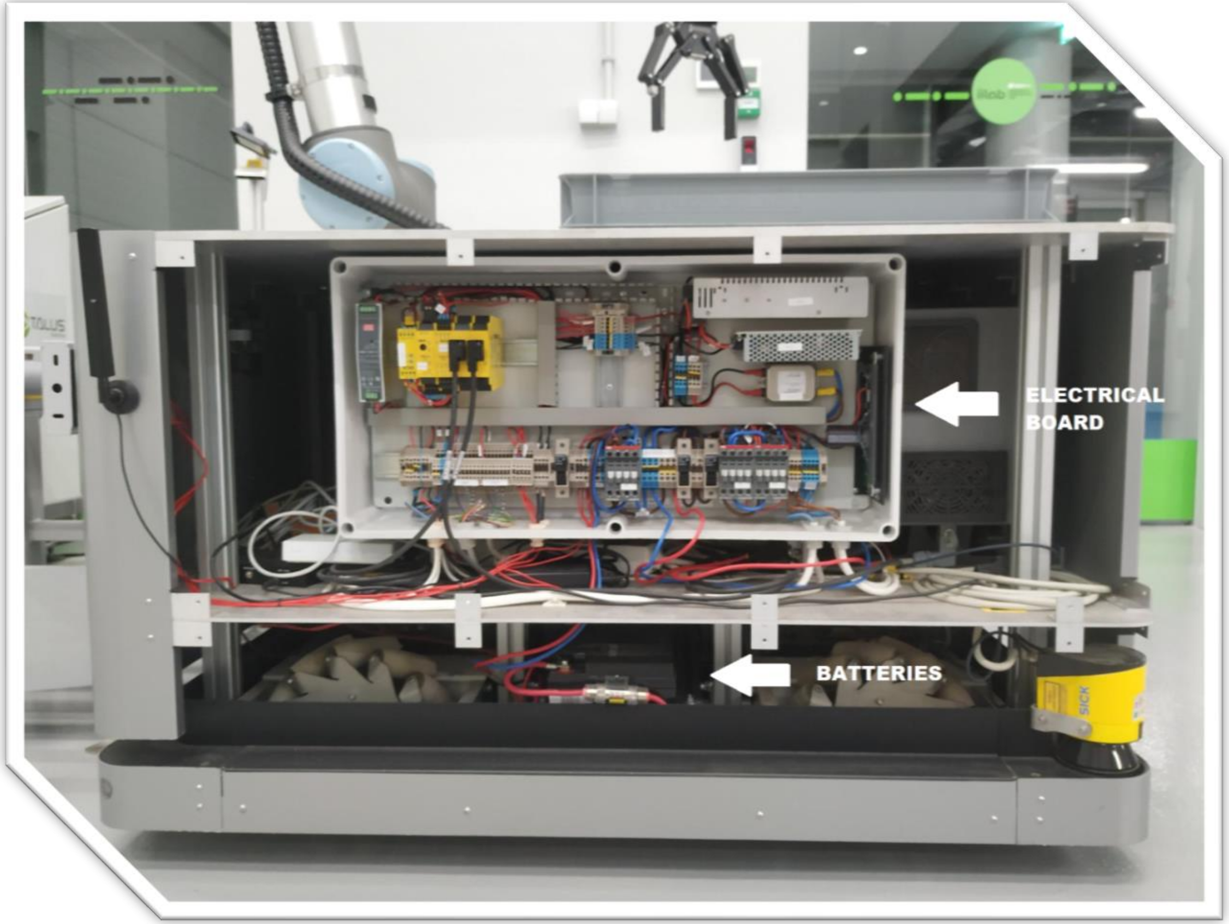
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

- Power System

- RESUME

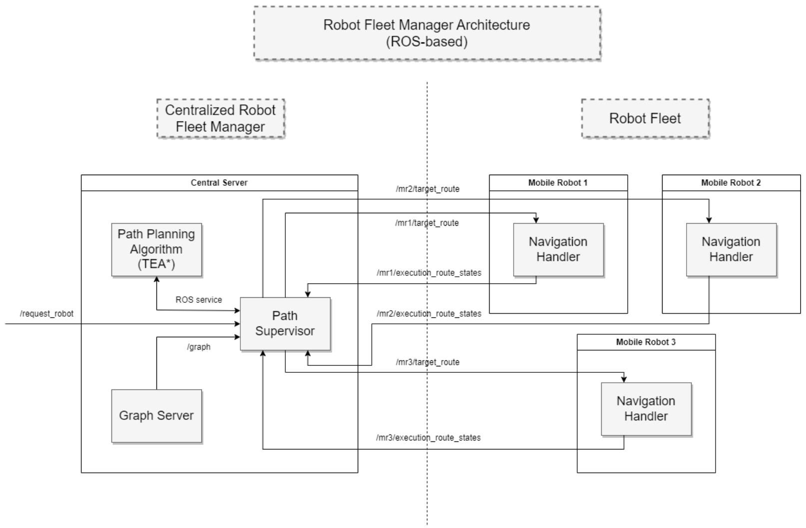
MARI4YARD
MARI4ALLIANCE

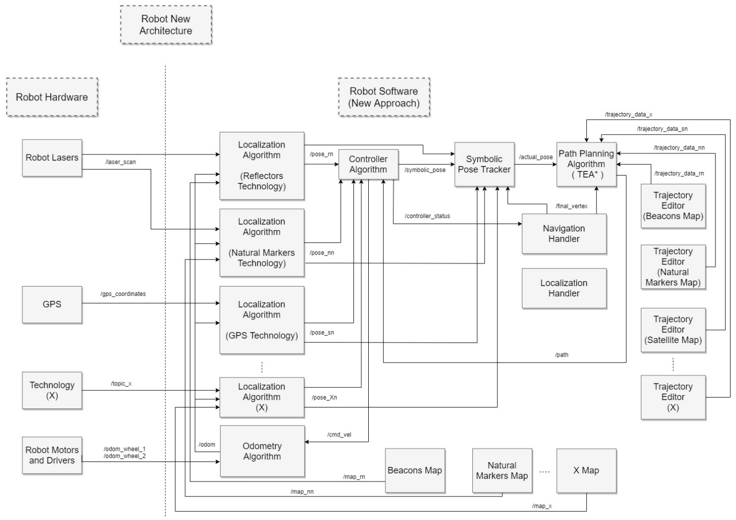


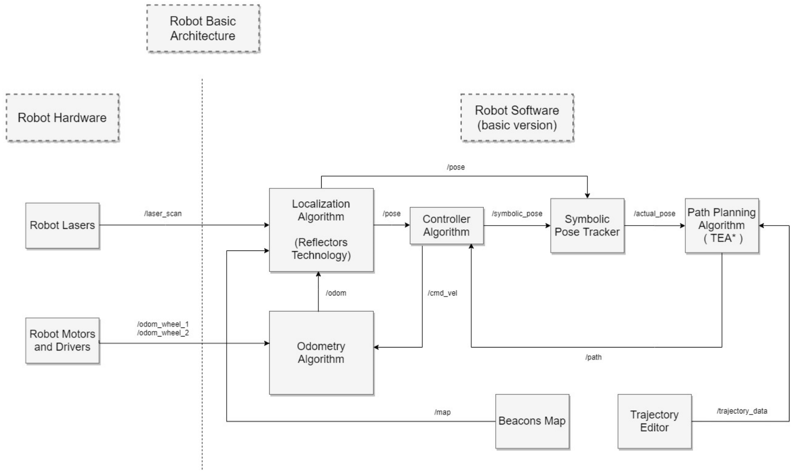
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Software







This project has received funding from
the European Union’s Horizon 2020
research and innovation programme
under grant agreement No 101006798

INESCTEC

ROBOT SOFTWARE - INDEX

A. Navigation Stack Architecture System

B. Navigation Stack Modules

1. Odometry Module
2. Localisation Module
3. Controller Module
4. Fleet Manager Module

C. Navigation Stack Installation and Configuration

1. ROS Workspace
2. Deb File – INESC TCE License Software
3. Robot Configuration Folders

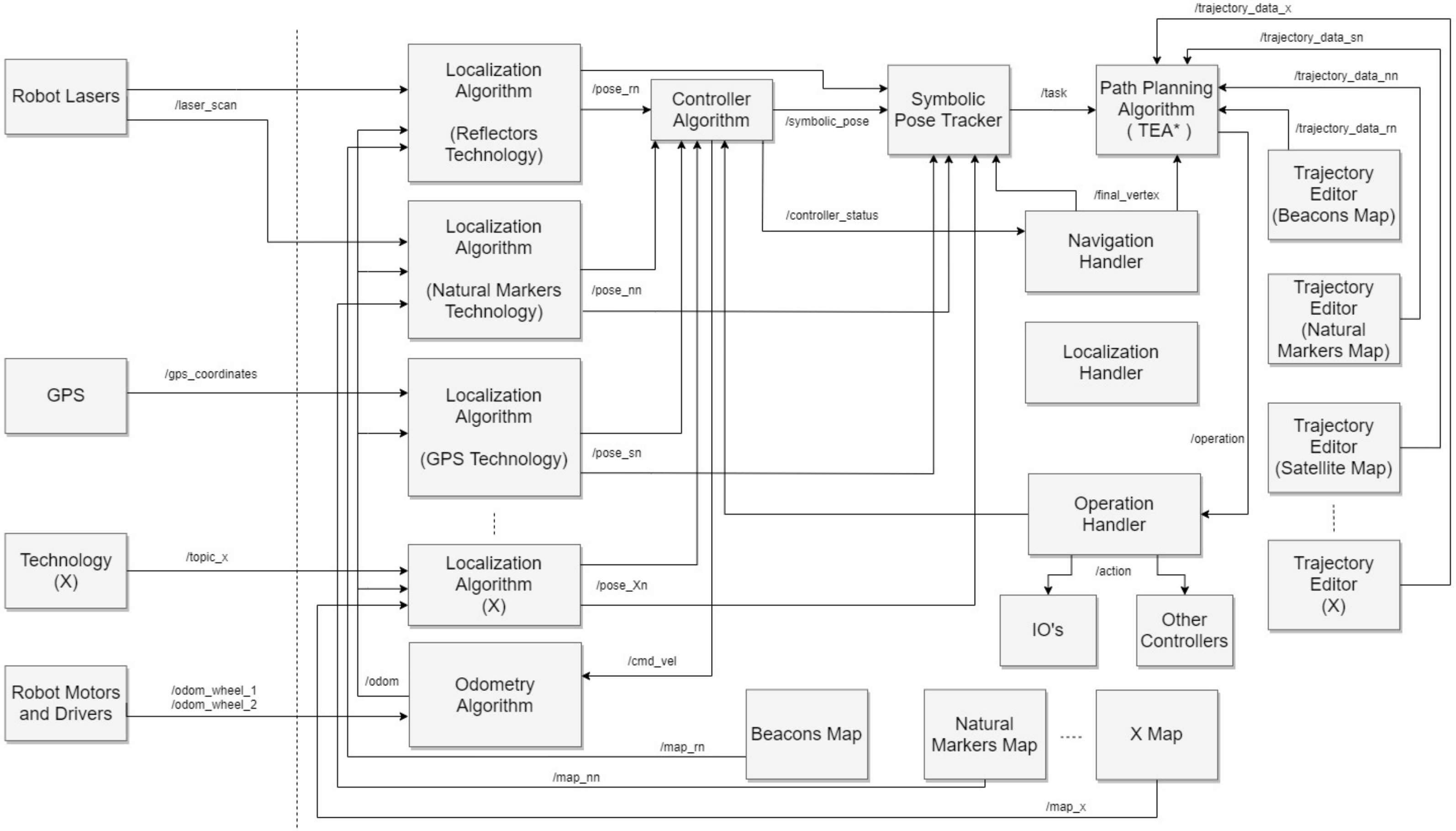
D. Human-Machine Interface Installation and Configuration

1. IRIS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robot Software

Navigation Stack Architecture System



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robot Software

Navigation Stack Modules

Localization Algorithm

Path Planning Algorithm

Odometry Algorithm

Controller Algorithm

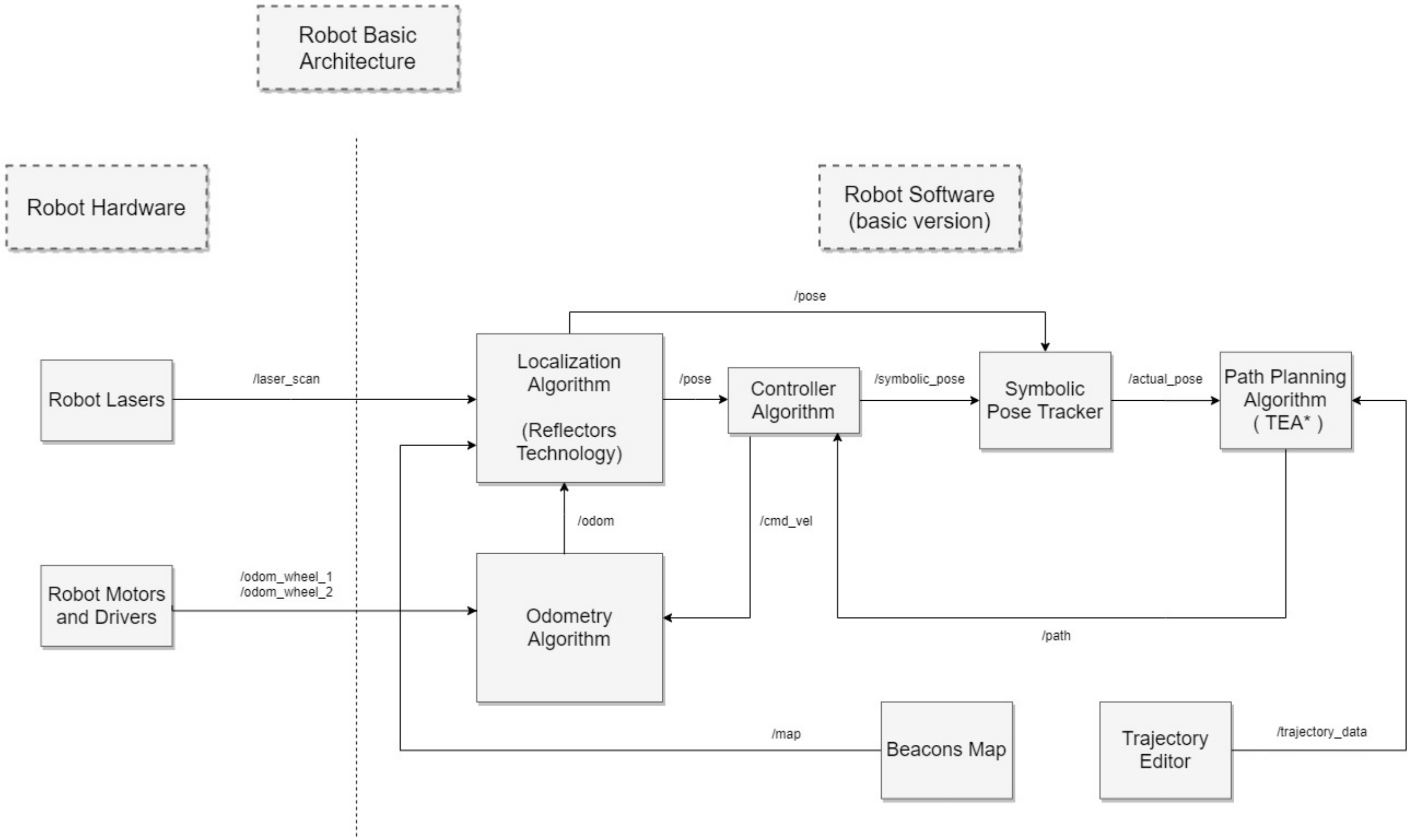
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Navigation Stack Modules

- Odometry

MARI4YARD
MARI4ALLIANCE

- Odometry is the use of data from motion sensors (drivers) and robot's wheels to estimate change in position over time;
- This data is based on encoder ticks and wheels parameters (diameter/perimeter);
- It is used in mobile robots to estimate their position relative to a starting location.

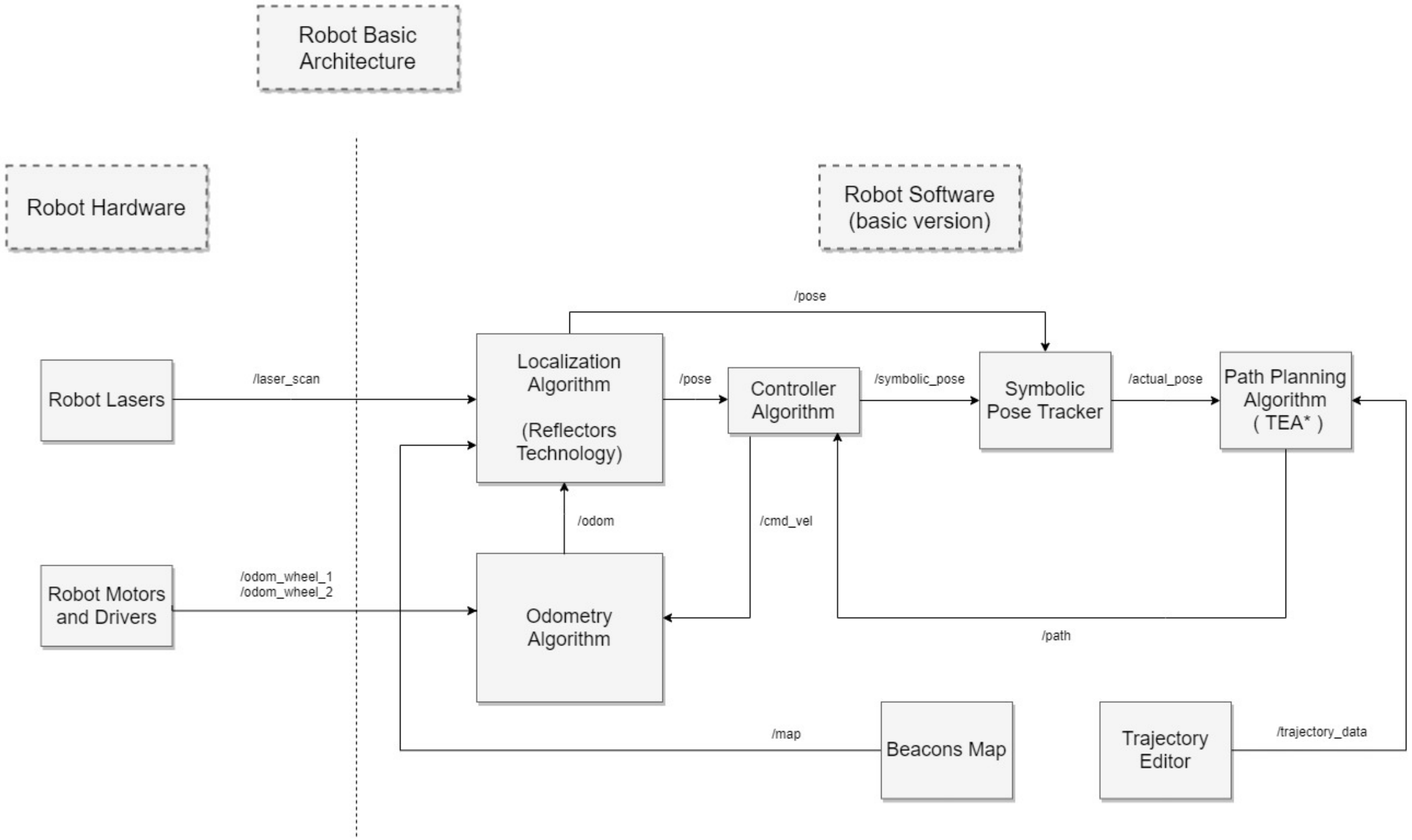


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Navigation Stack Modules

- Controller



- Proportional Integral Derivative (PID) controllers are employed for trajectory tracking and speed control;
- This ensures that the robot follows desired paths accurately and maintains desired speeds during navigation;
- Navigation Stack Controllers Types: Differential, Tricycle, Omnidirectional.

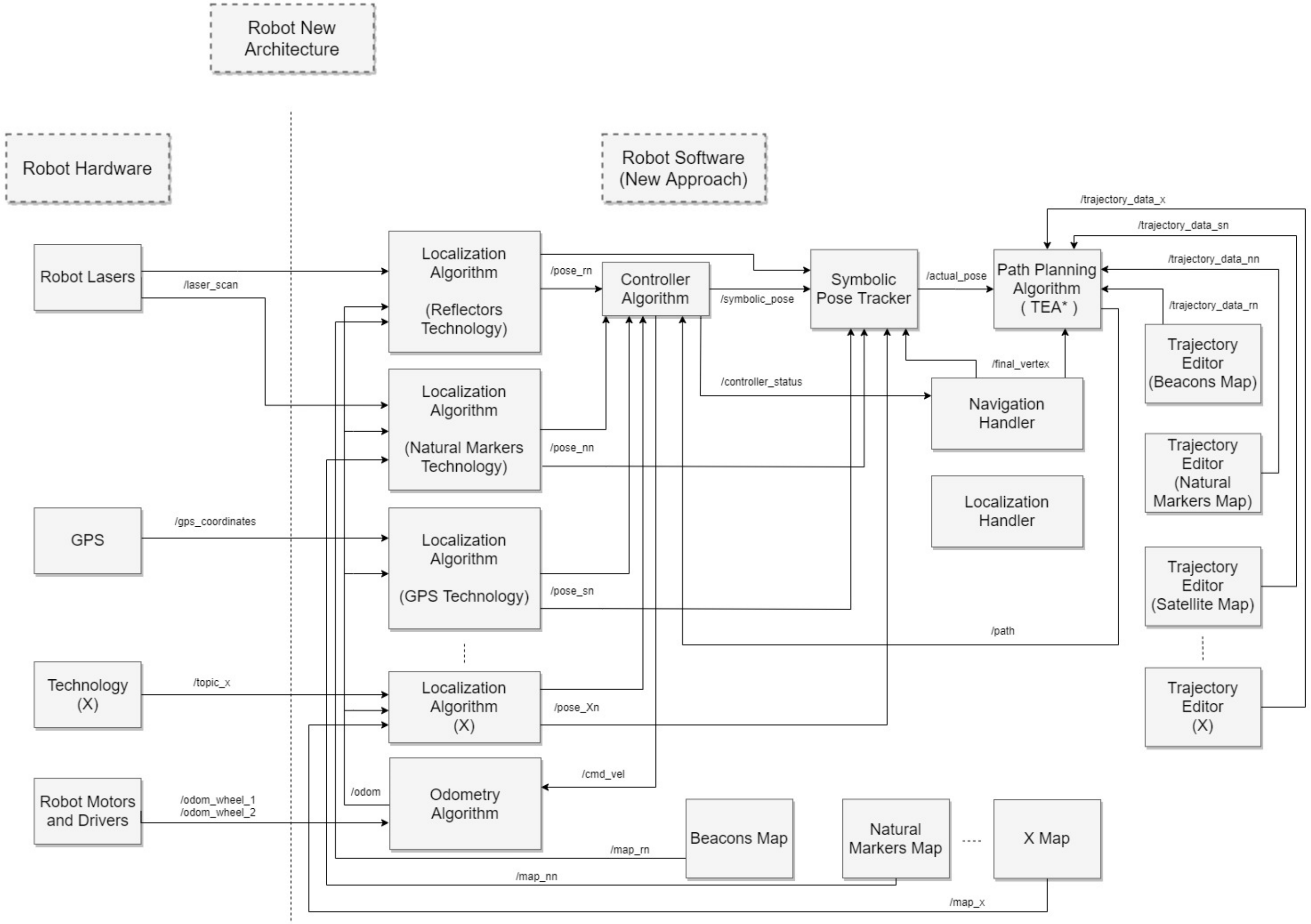
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Navigation Stack Modules

- Localization

- Natural Markers/Contours - Localization Perfect Match Algorithm;
- Beacons Map - Extended Kalman Filter Beacons Algorithm (used for docking and precise movements)
- Satellite Map – GPS Technology Algorithm
- Localization Handler – responsible for switching, in specific waypoints, between different localization algorithms, maps and trajectories;



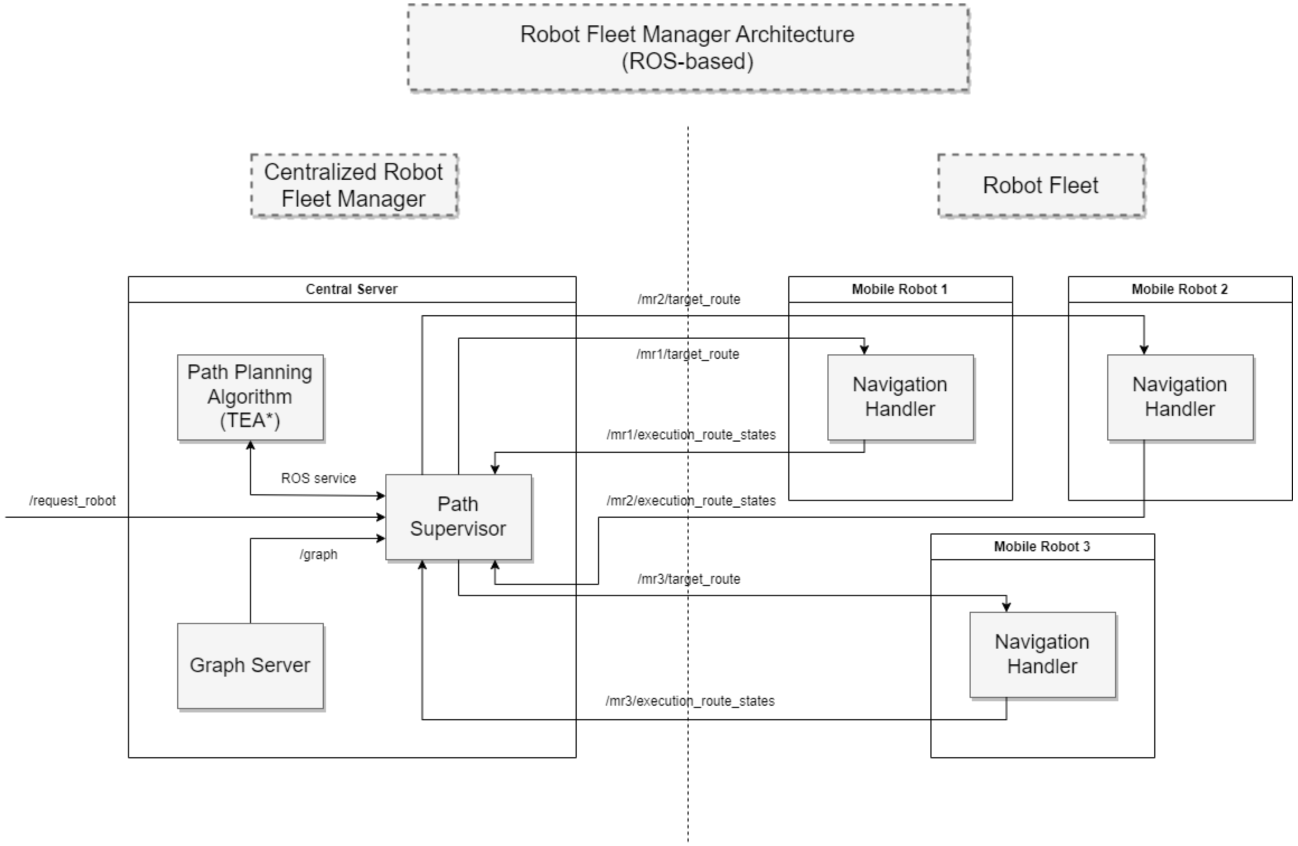
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Navigation Stack Modules

- Fleet Manager

MARI4YARD
MARI4ALLIANCE

- **Path Planner (TEA*)** - consider the problem of finding optimal paths using local information and ensuring that the robot is not lost;
- **Path Supervisor** - responsible for supervising, in real time, all the paths dimensioned by the path planner for all the robots. If any robot is not complying with the plan, the supervisor stops the robots and asks the path planner for new trajectories for the robots. This avoids collisions and deadlocks;
- **Graph Server** - is executed only once and its purpose is to load the graph from the system to the supervisor.

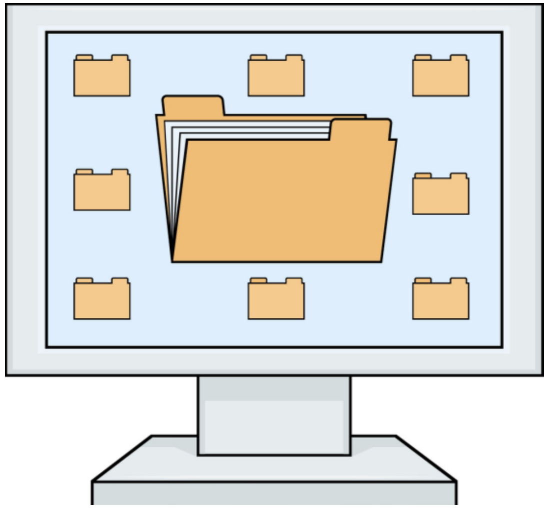


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Software

Navigation Stack Installation and Configuration



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

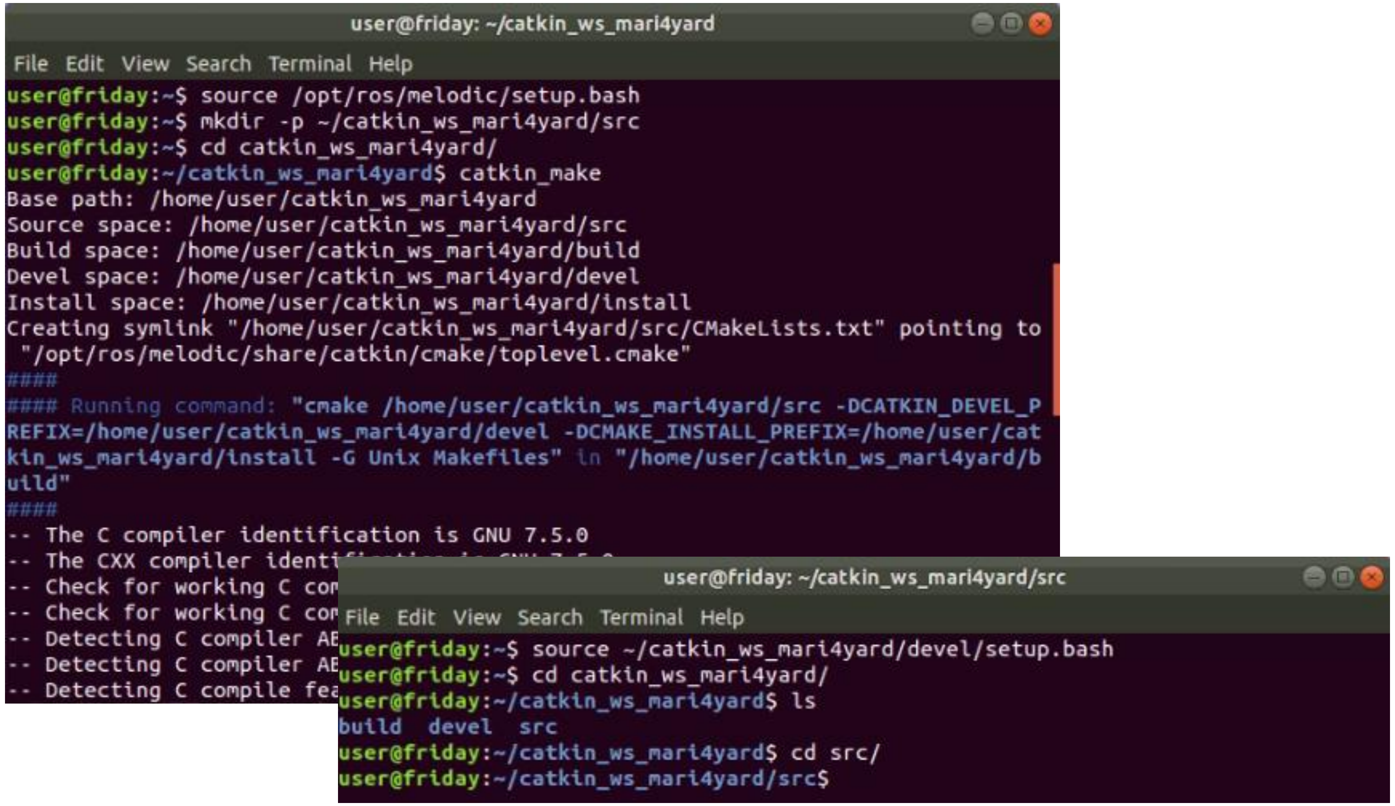
• Navigation Stack Installation and Configuration

- ROS Workspace

MARI4YARD
MARI4ALLIANCE

- Ubuntu 18 (LTS* Version) – ROS Melodic
- Ubuntu 20 (LTS* version) – ROS Noetic

** LTS – Long Term Support*



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Navigation Stack Installation and Configuration

- Deb File – INESC TEC License Software

MARI4YARD
MARI4ALLIANCE

- CRIIS Redmine Repository
- Navigation Stack Execution File

Navigation Stack Inesctec Robotics Releases

+ Overview Activity Roadmap Issues Wiki Settings

Navigation Stack Inesctec Robotics Releases

Mobile Robotics Introduction Guide

1. Drive INESC TEC: https://drive.inesctec.pt/s/kLprEGxm2rae6ZY

Sentinel License Pens List

1. Drive INESC TEC: https://drive.inesctec.pt/s/nW7JHSjGwXET3r2

Installation Notes

Prerequisites:

1. Ubuntu 18, AMD64
2. Updated operating system
3. ROS instaled: http://wiki.ros.org/ROS/Installation

Installation:

1. Download and install deb file: aksusbd_7.90-1_amd64.deb (run-time environment for Sentinel LDK deb file):
   - aksusbd_*_amd64.deb
2. Download and install Navigation Stack deb file:
   - jarvis-system-melodic_*.deb -> **Ubuntu 18.04 - 64 bits**

Obs:

- The .deb files are installed using the following command: $sudo apt install ./jarvis-system-melodic_0.7-9.deb
- If there are dependencies missing, try to update your system:
  - $sudo apt update
  - $sudo apt dist-upgrade

Test:

1. Don't forget your license key (It is always necessary to have the pen inserted in the PC while using the Navigation Stack)
2. Don't forget to load your environment variables:
   - $source /opt/ros/$ROS_DISTRO/setup.bash
3. Run the following command:
   - $roslaunch jarvis_nav_conf wake_up_great_jarvis.launch
   - **You need to have graphical support to run this command**

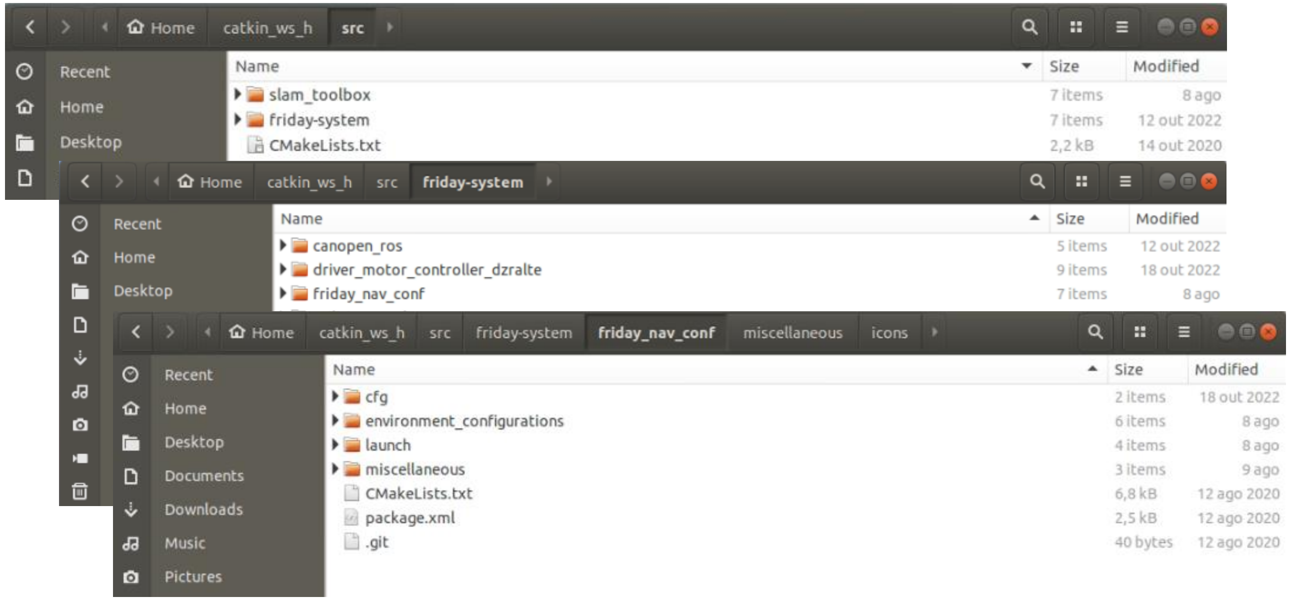
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

- Navigation Stack Installation and Configuration

- Robot Configuration Folder – Robot_Nav_Conf

MARI4YARD
MARI4ALLIANCE

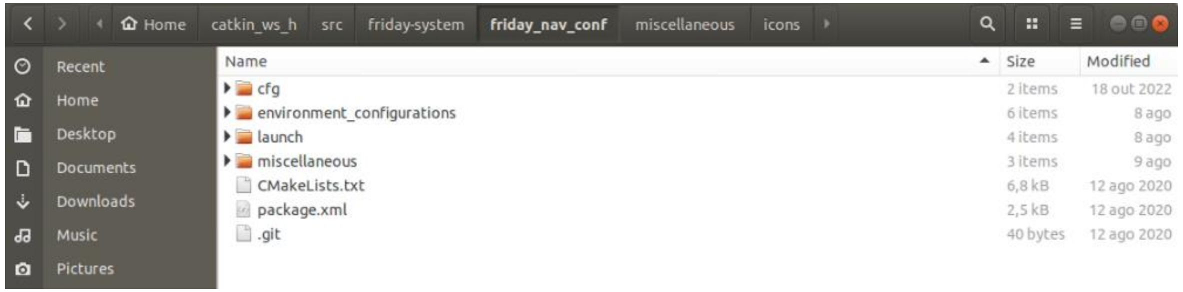


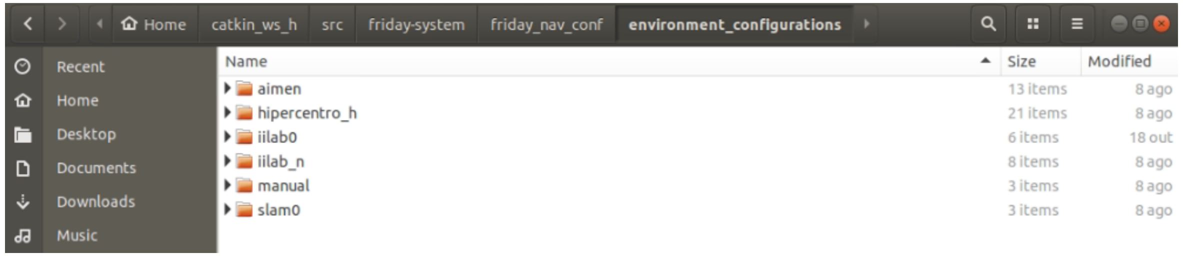
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

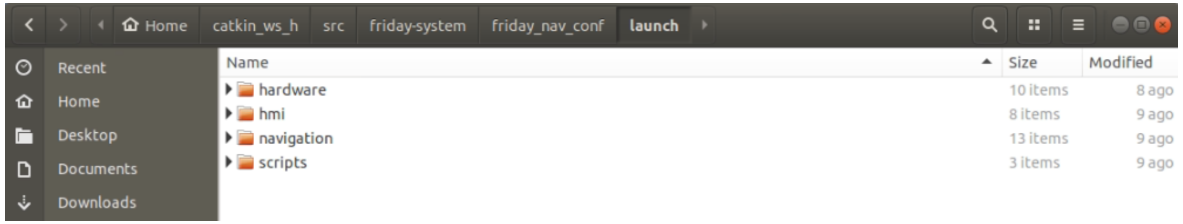
INESCTEC

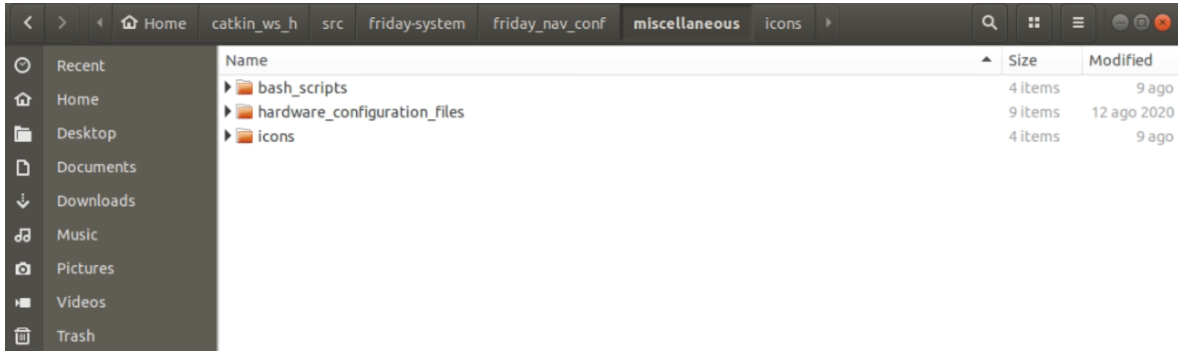
• Navigation Stack Installation and Configuration

• Friday_Nav_Conf Folder - Architecture









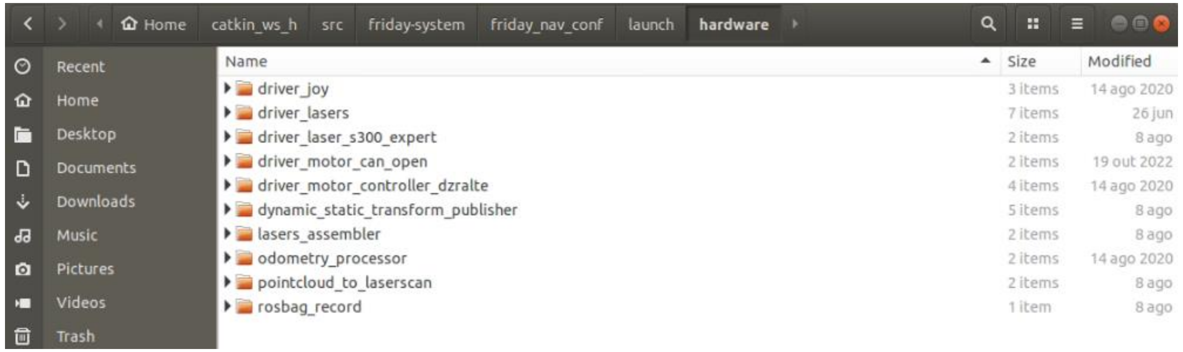
MARI4YARD
MARI4ALLIANCE

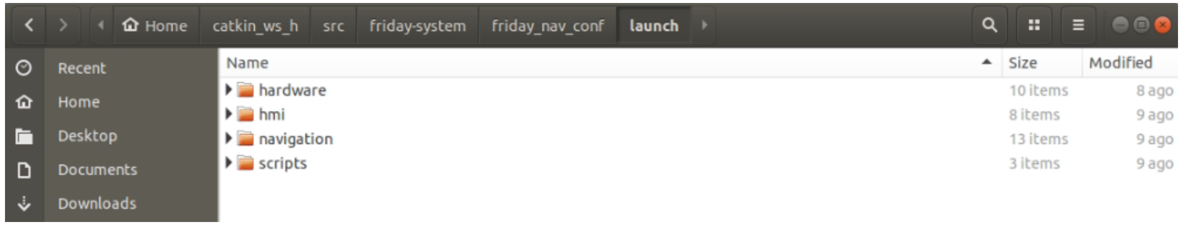
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

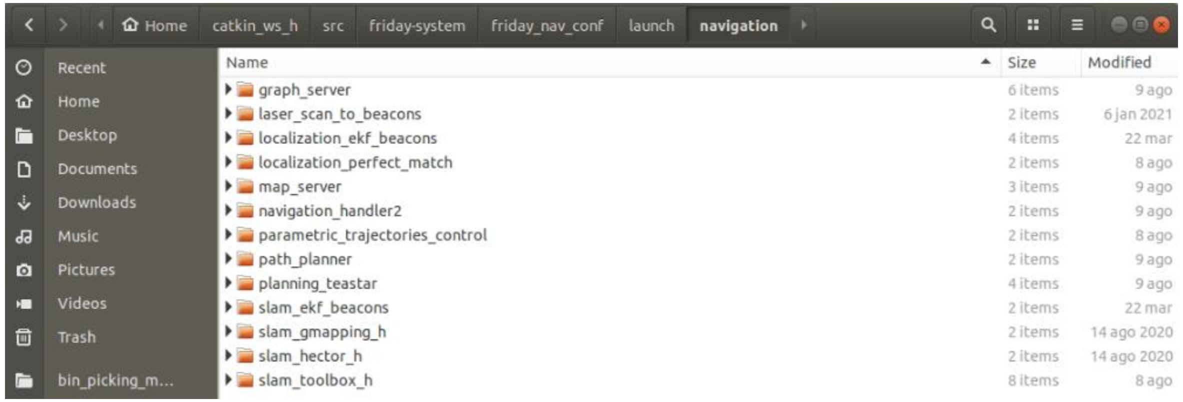
INESCTEC

• Navigation Stack Installation and Configuration

- Friday_Nav_Conf – Launch Folder







MARI4YARD
MARI4ALLIANCE

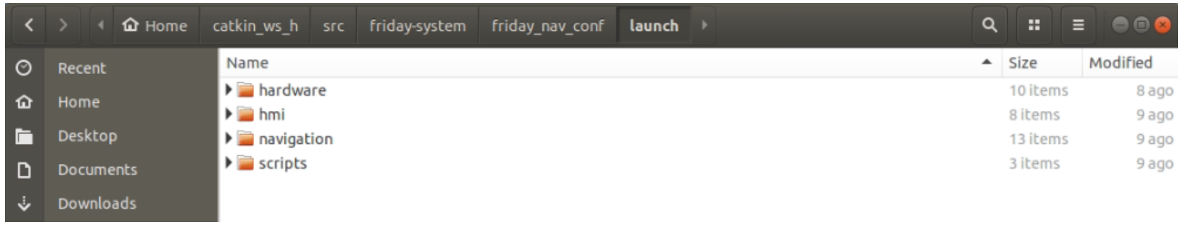
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

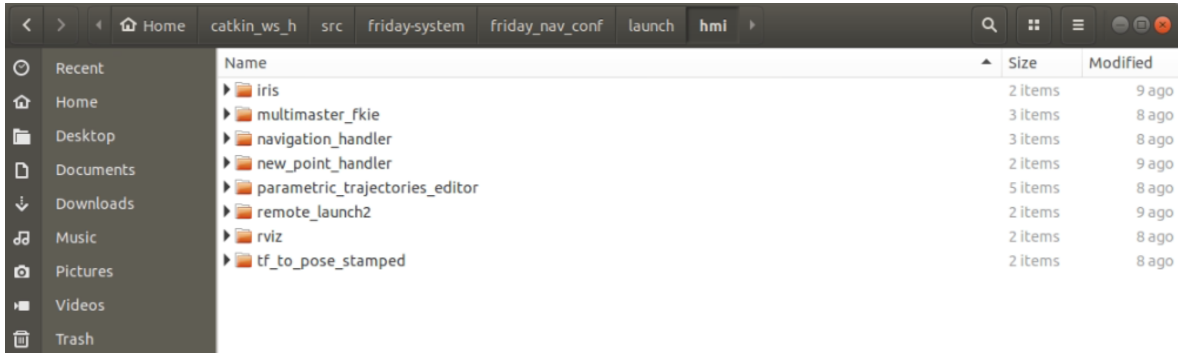
INESCTEC

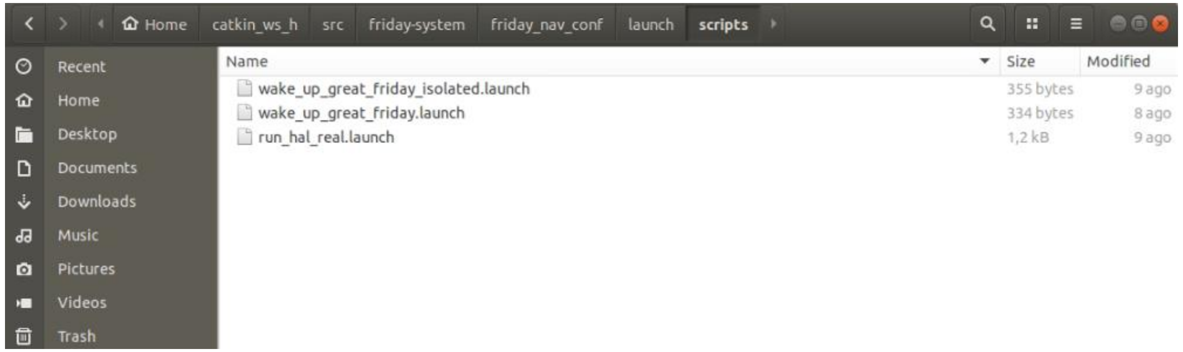
• Navigation Stack Installation and Configuration

- Friday_Nav_Conf – Launch Folder

MARI4YARD
MARI4ALLIANCE







This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Software

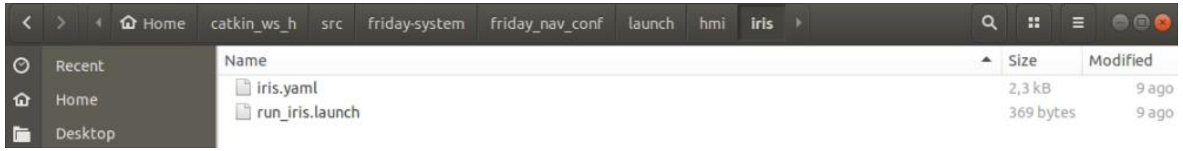
Human-Machine Interface Installation and Configuration

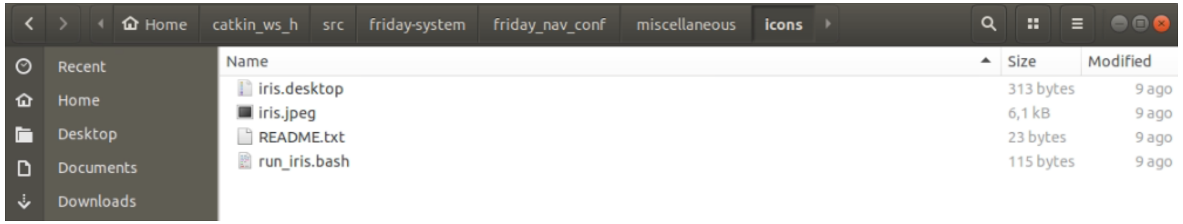
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• Friday_Nav_Conf – IRIS Installation

MARI4YARD
MARI4ALLIANCE





- Edit the iris.desktop file with the correct directories for the parameter Exec and Icon;
- Move the edited file to the Autostart Ubuntu's folder.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Robot Configuration

IRIS

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robot Configuration (IRIS) - INDEX

A. Environment Application

B. Mapping Operation

C. Navigation Operation

D. Move Robot

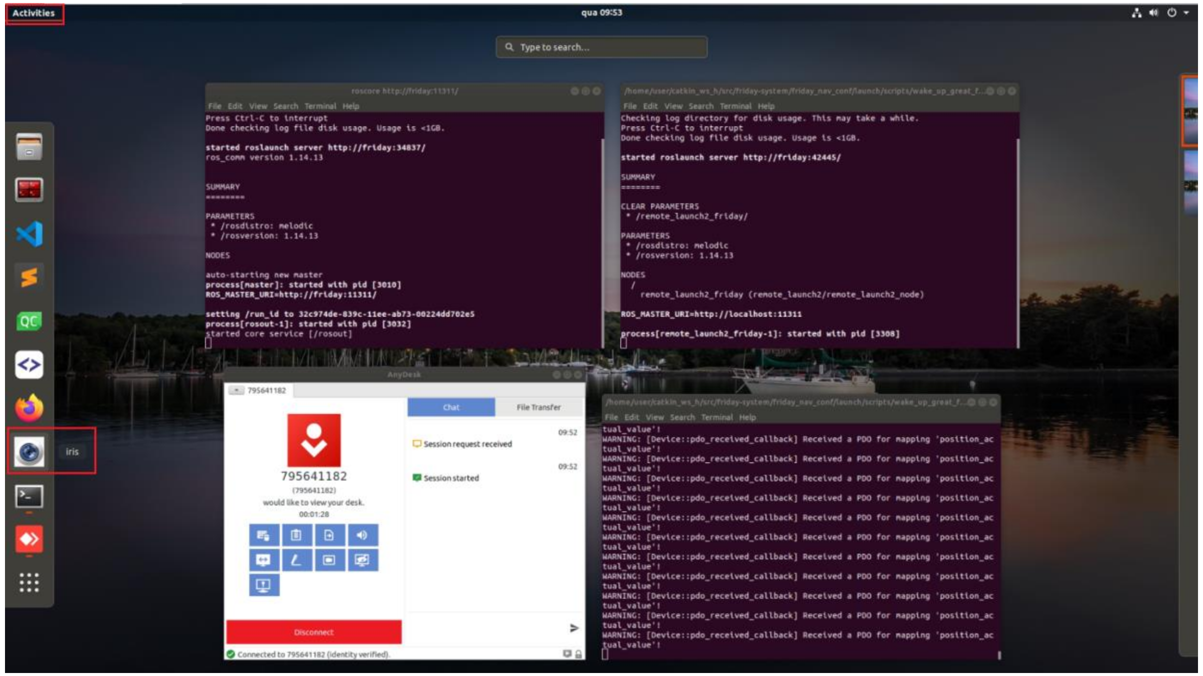
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration

1. Click on ***Activities*** in the left top screen corner;

2. The left bar will appear. After Click on ***IRIS Application***;



MARI4YARD
MARI4ALLIANCE

- This interface allows the configuration of the navigation system for a new installation or configuration;
- It is where the routes, used by the mobile robots, are mapped and defined.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

The mapping operation is a fundamental part of the system's initial configuration. It consists on the creation of an occupancy matrix that represents the system's operation space.

This matrix delimits the space in which the system can operate and where possible obstacles are located.

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

3. Click on the ***Mapping Tab*** to start, autonomously, the mapping operation;

4. Use the joystick to move the mobile platform in order to visit all the areas in which the system will operate;

During the mapping operation, IRIS displays the generated map in real-time;

5. Once the mapping process is completed, Click on ***Save Map*** button to save the robot's map;

6. If there are more floors to map, move the robot to the new floor, click ***Reset*** and repeat the previous steps.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

wait until you see something identical to the image on the right on the screen

• Human-Machine Interface Installation and Configuration

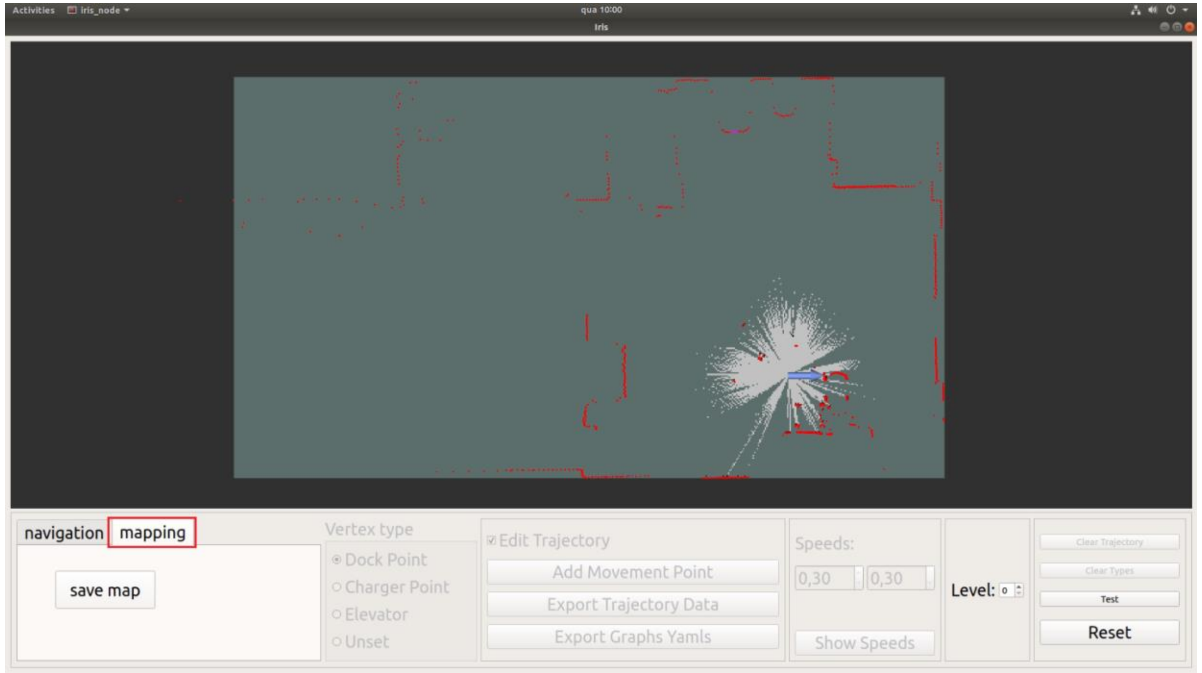
• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

3. Click on the ***Mapping Tab*** to start, autonomously, the mapping operation;

WAIT 2 MINUTES! BE PACIENT!

The system needs time to launch some new packages.



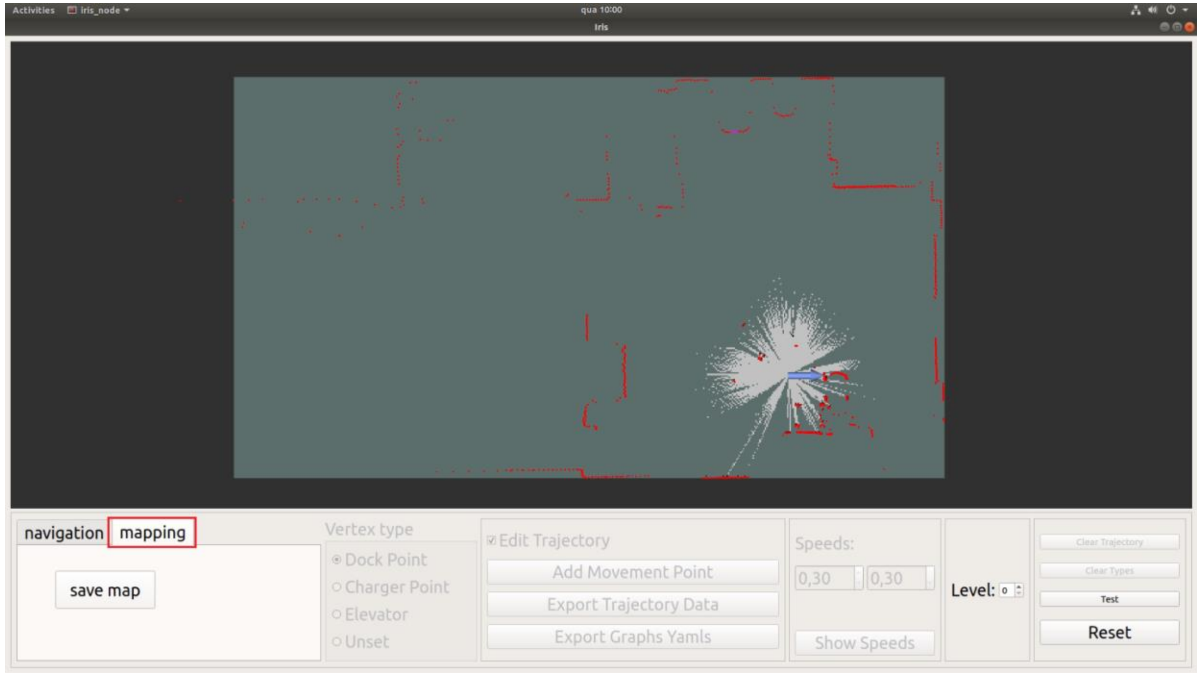
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

wait until you see something identical to the image on the right on the screen

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation



3. After getting some data displayed on the screen make the following steps:

- ***Mouse Scroll*** for Zoom Out or Zoom In;
- ***Hold*** the ***Middle Button*** and drag/center the image.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

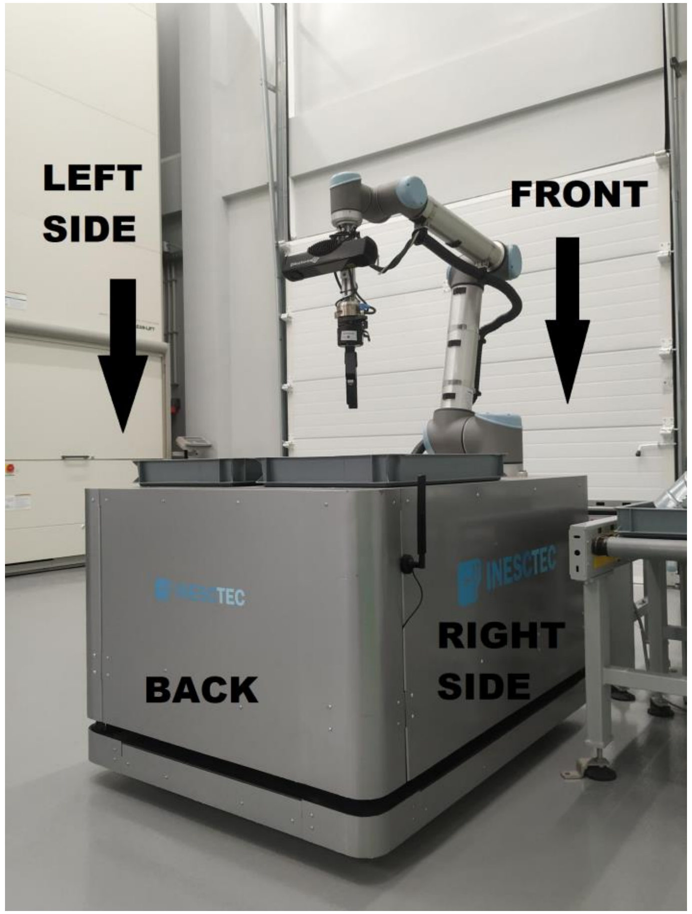
Human-Machine Interface Installation and Configuration

- IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

PAY ATTENTION TO THE ROBOT'S ORIENTATION!!

Before Move the Robot



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

1. ***Hold Down*** the ***LB*** button to move the robot;

MARI4YARD
MARI4ALLIANCE

4. Use the joystick to move the mobile platform in order to visit all the areas in which the system will operate.

BE CAREFUL! BE PRUDENT!

During the mapping operation, IRIS displays the generated map in real-time;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Mapping Operation

2. The ***Left Analogue*** button is used to control the robot forwards, backwards, left and right;

BE CAREFUL! BE PRUDENT!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

3. The ***Right Analogue*** button is reserved for the rotation movement.

BE CAREFUL! BE PRUDENT!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

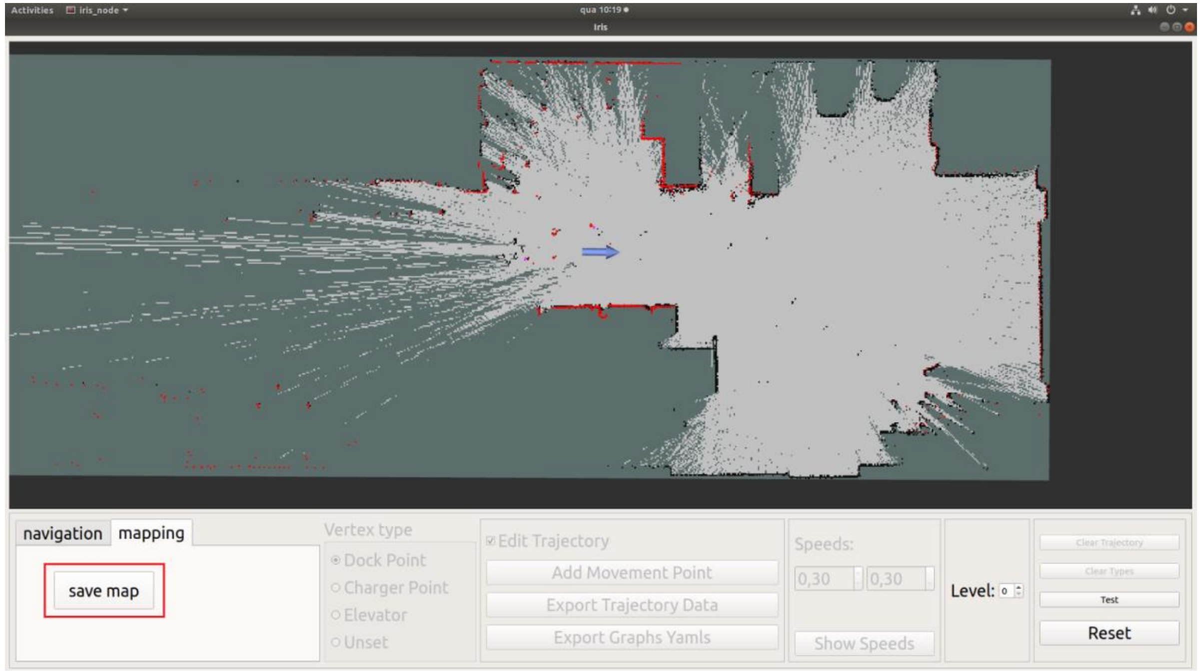
• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

5. Once the mapping process is completed, Click on ***Save Map*** button to save the robot's map.

WAIT 1 MINUTE! BE PACIENT!

The system takes time to save the new files.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

The navigation operation consists on the creation of a new trajectory for the robot and finally move the robot autonomously.

Edges Types:

- Unidirectional – Robot moves only in one direction;
- Bidirectional – Robot moves in two directions;

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

1. Click on the ***Navigation Tab*** to switch from the previous operation. The system launch, autonomously, the latest created map;

2. Click on the ***Clear Trajectory*** button and then in ***Clear Types*** button to erase the latest stored trajectory. The system create just one vertex in the origin map position;

3. After is necessary locate the mobile platform in the new map. Click on the ***Set Pose*** button and follow the next steps:
   - Move mouse to the supposed robot's position;
   - Hold the Left Mouse Click and then drag it in the orientation the robot is in;
   - If you find that the robot is not located on the new map, repeat the steps again.

4. After located, add the second vertex on robot's pose. Click on the ***Add Movement Point*** button. The system will create a new vertex with the robot's orientation;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

The navigation operation consists on the creation of a new trajectory for the robot and finally move the robot autonomously.

Edges Types:

- Unidirectional – Robot moves only in one direction;
- Bidirectional – Robot moves in two directions;

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

5. Build the path by creating and configuring new vertices;

6. Edit the path by creating and configuring the edges between vertices;

7. Once the path configuration process is completed, Click on ***Export Trajectory Data*** button and on the ***Export Graph Yamls*** button to save the robot's trajectory;

8. After these steps, click on the ***RESET*** button and check that the robot is located and that the new map and trajectory are launched autonomously and correctly.

9. If so, it is possible to move the robot by right-clicking on the vertex you want to move the robot to and clicking on the ***Come Here*** button.

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

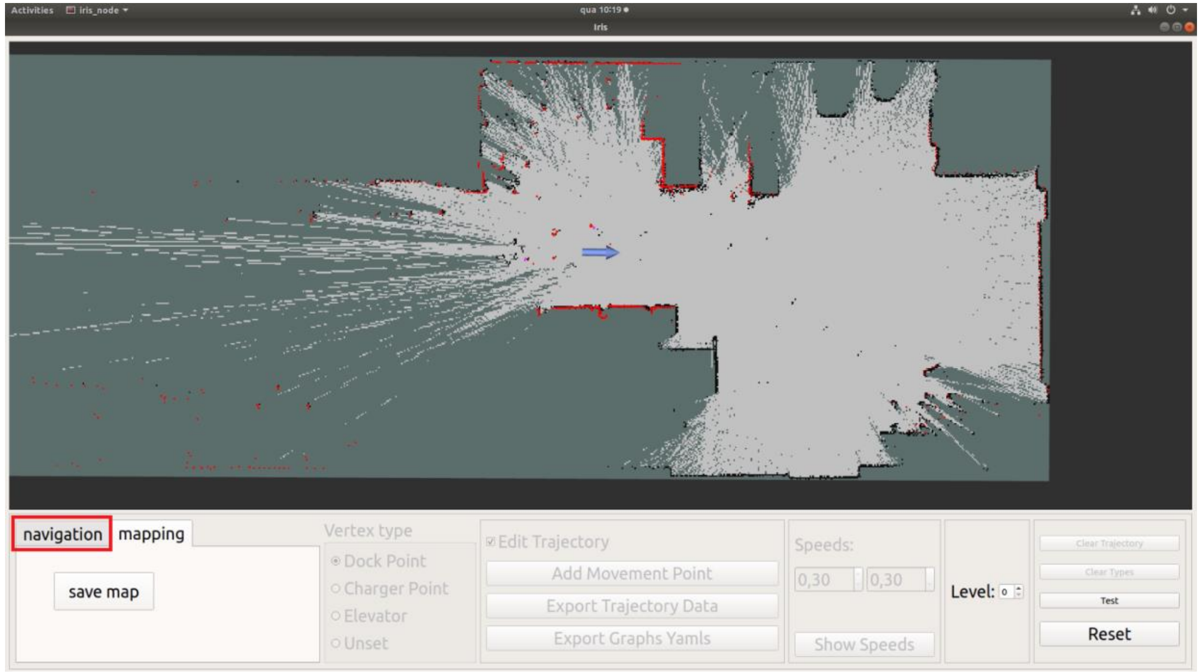
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation

1. Click on the ***Navigation Tab*** to switch from the previous operation. The system launch, autonomously, the latest created map;

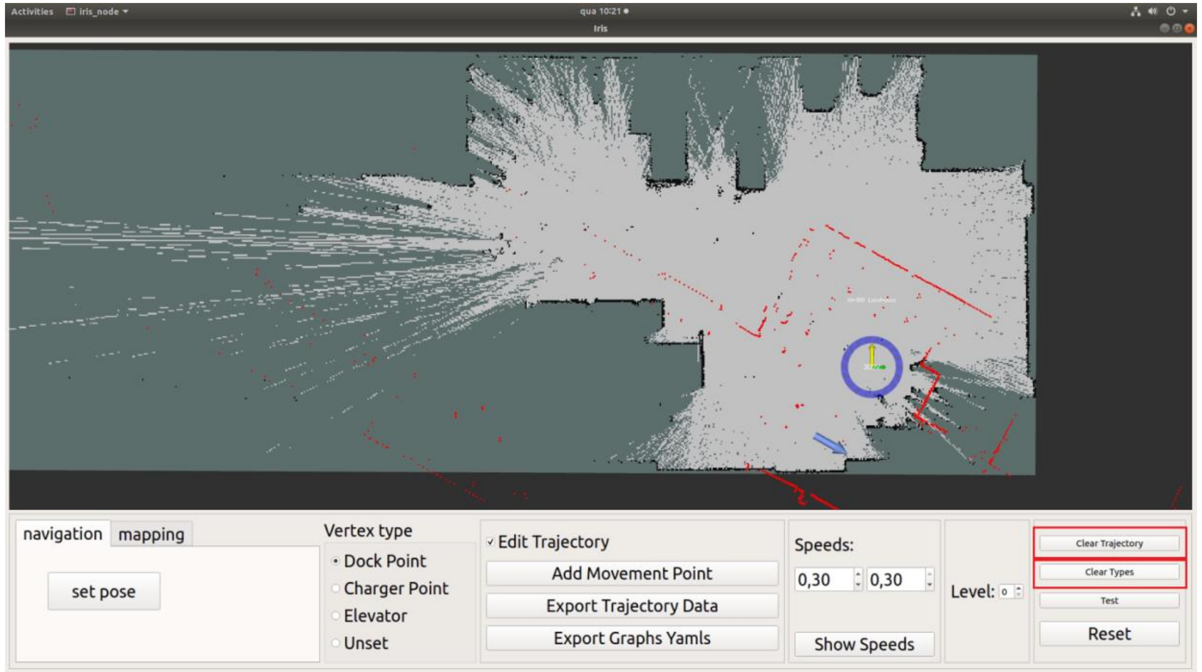


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

- Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation



2. Click on the ***Clear Trajectory*** button and then in ***Clear Types*** button to erase the latest stored trajectory. The system create just one vertex in the origin map position;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

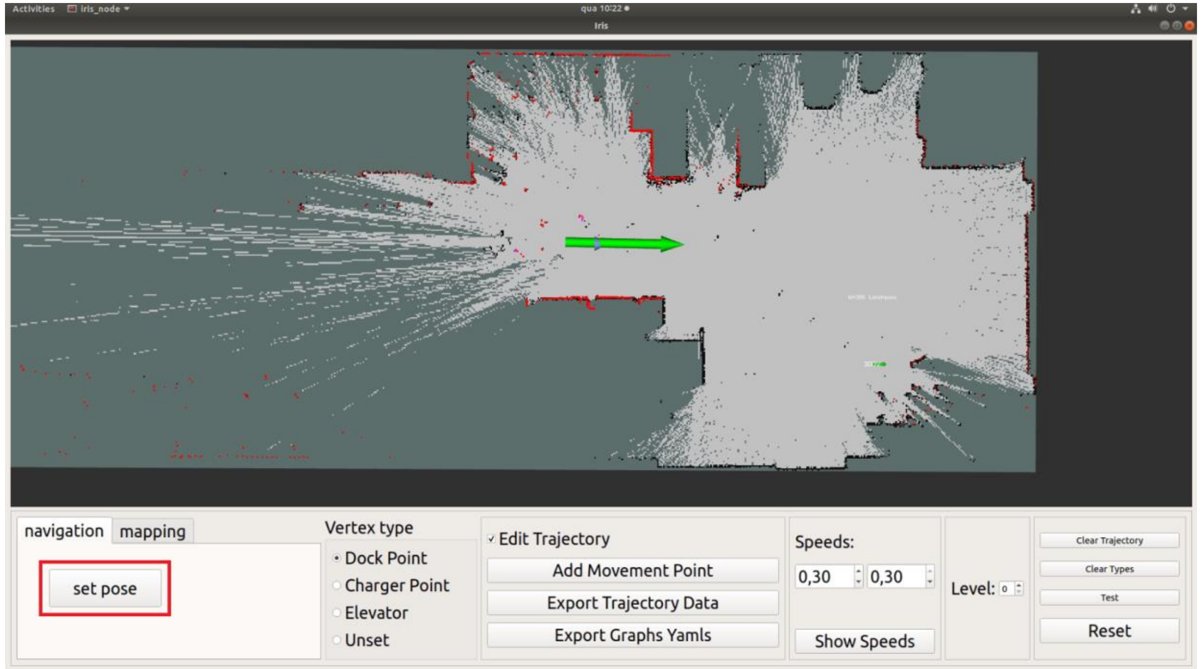
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

3. After is necessary locate the mobile platform in the new map. Click on the ***Set Pose*** button and follow the next steps:

- Move mouse to the supposed robot's position;
- ***Hold*** the ***Left Mouse*** Click and then drag it in the orientation the robot is in;
- If you find that the robot is not located on the new map, repeat the steps again.

NOTE: The robot is located only when the red lasers points match with the black wall points on the new map.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

4. After located, add the second vertex on robot’s pose. Click on the ***Add Movement Point*** button. The system will create a new vertex with the robot’s orientation;

NOTE: Inside the green rectangle is possible to see the new vertex and the robot located. The red points are the real-time lasers points data.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

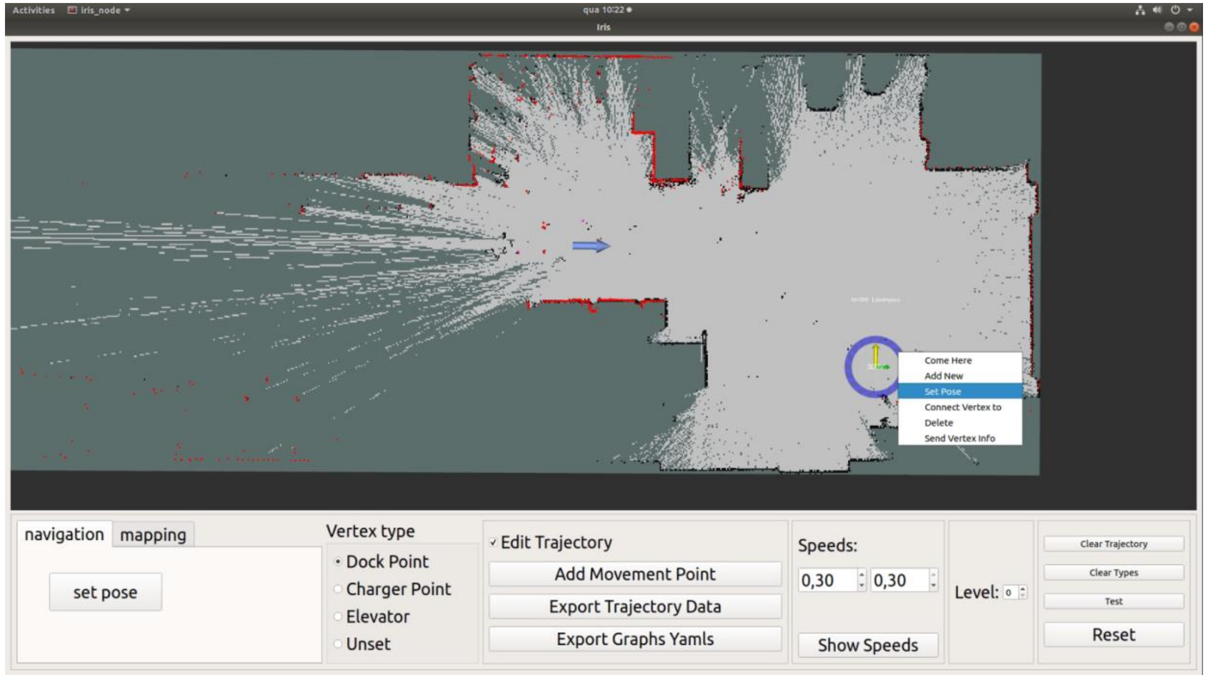
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

5. Build the path by creating and configuring new vertices, following the next steps:

- ***Right Mouse Click*** on the blue circle of a vertex;
- Click on ***Add New***;
- ***Hold*** the ***Left Mouse Click*** on the green arrow, on the created vertex, and move it to the desired map point;
- ***Hold*** the ***Left Mouse Click*** on the blue circle of the moved vertex and rotate the circle until you get the direction of the desired trajectory, represented by the green arrow direction;
- Finally, adjust the yellow arrow alluding to the robot's orientation at that waypoint/vertex by ***Holding*** the ***Left Mouse Click*** on the yellow arrow and rotating it until the desired orientation is obtained.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

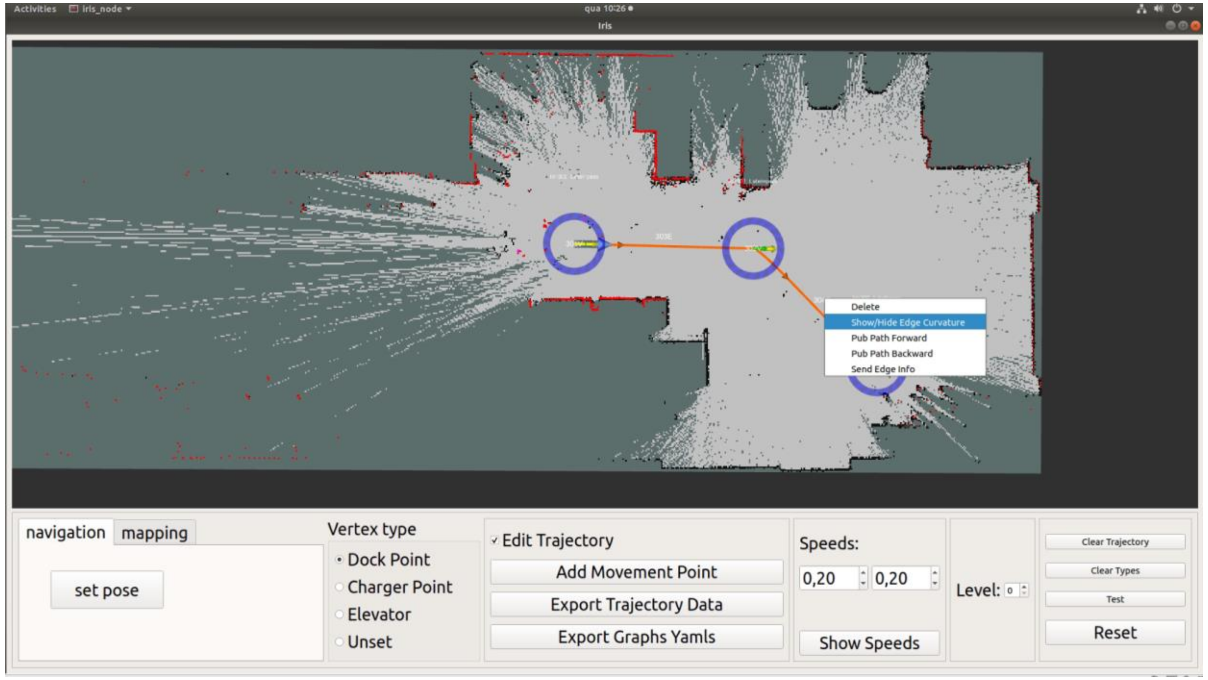
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

6. Edit the path by creating and configuring the edges between vertices, following the next steps:

- ***Right Mouse Click*** on the blue circle of a vertex;
- Click on ***Connect Vertex To***;
- ***Left Mouse Click*** on the blue circle of the second vertex - it is essential that the two vertices do not have opposite orientations;
- The new Edge is created between the two defined vertices;
- ***Right Mouse Click*** on the orange line of an edge to adjust it;
- Click on ***Show/Hide Edge Corvature***. (**See Next Slide)**



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

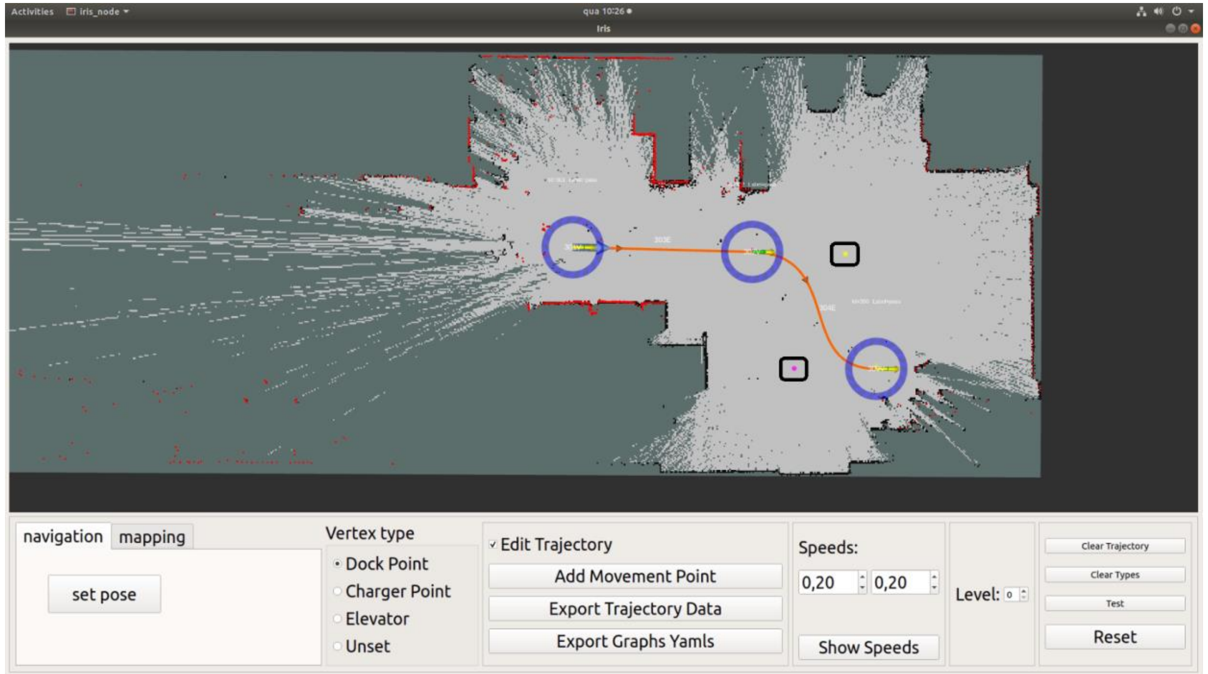
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

6. Edit the path by creating and configuring the edges between vertices, following the next steps:

- ***Holding*** the ***Left Mouse Button*** adjust, sliding, the yellow and the pink dots - the aim is to make the curve as smooth as possible;
- ***Right Mouse Click*** on the same orange line of the respective edge;
- Click on ***Show/Hide Edge Corvature*** for the dots to disappear from the image.



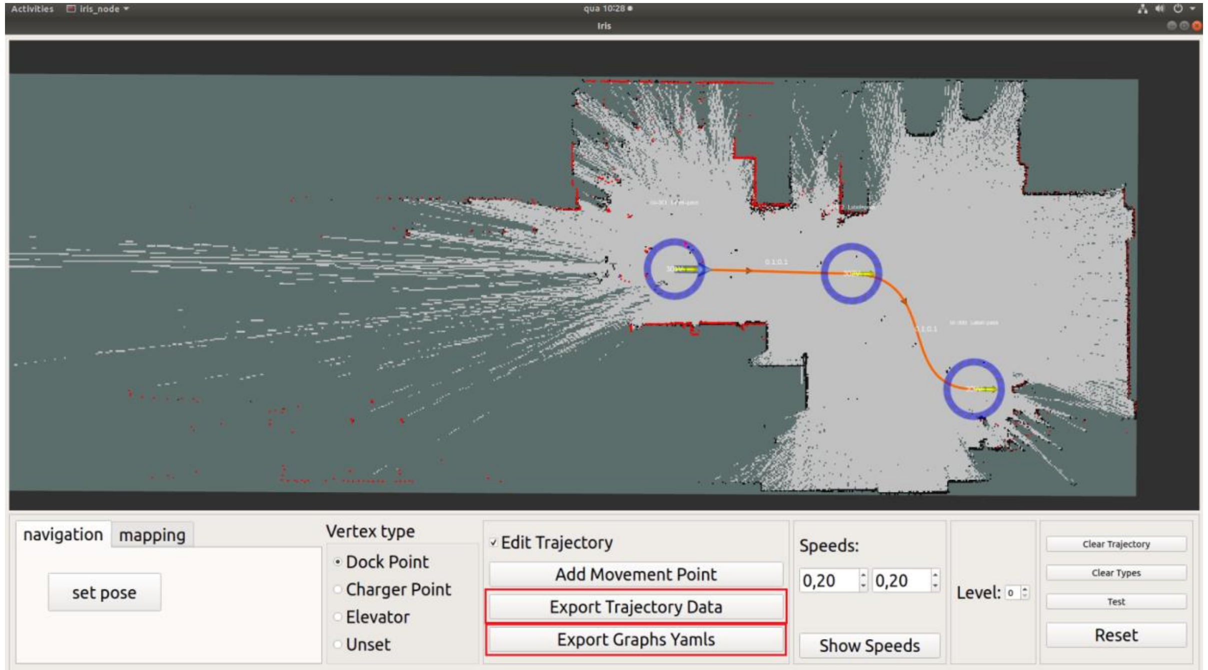
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

7. Once the path configuration process is completed, Click on ***Export Trajectory Data*** button and on the ***Export Graph Yamls*** button to save the robot's trajectory;

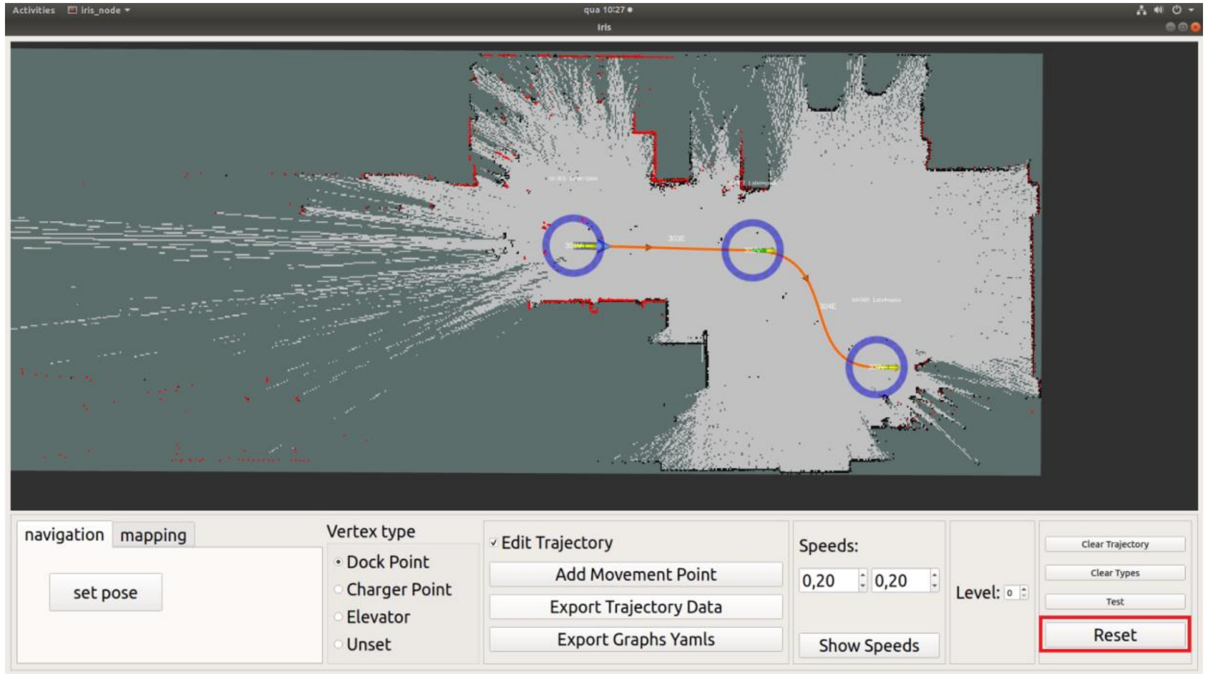


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

8. After these steps, click on the ***RESET*** button and check that the robot is located and that the new map and trajectory are launched autonomously and correctly.



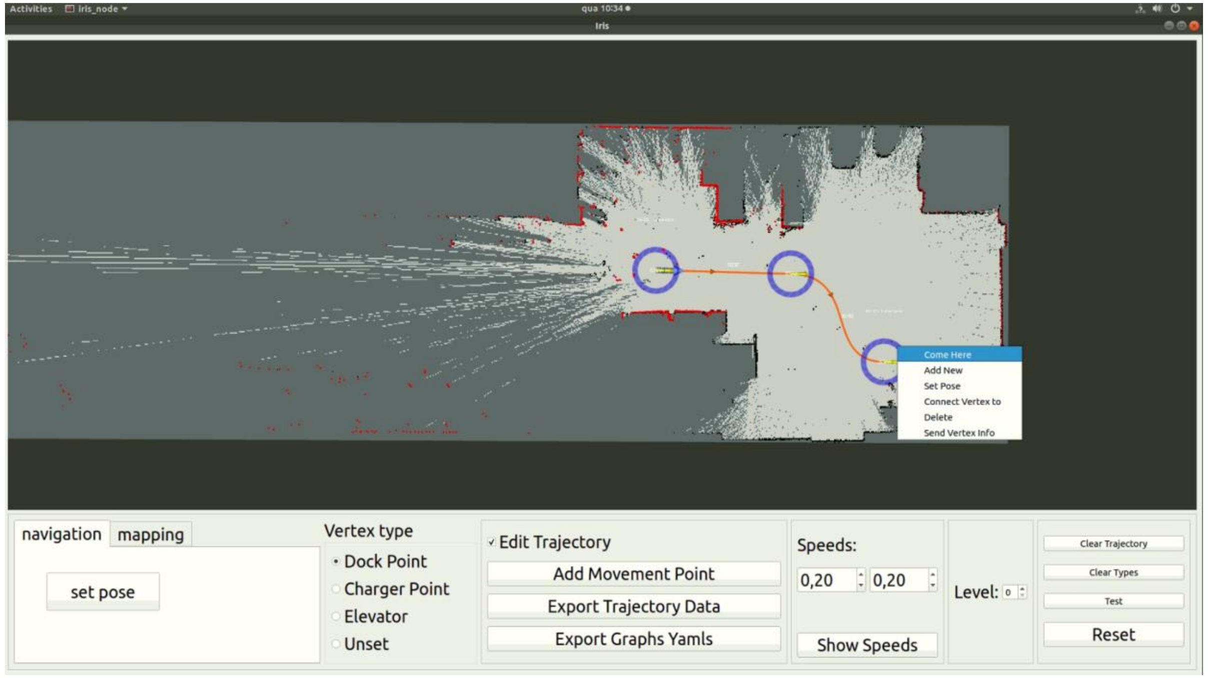
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

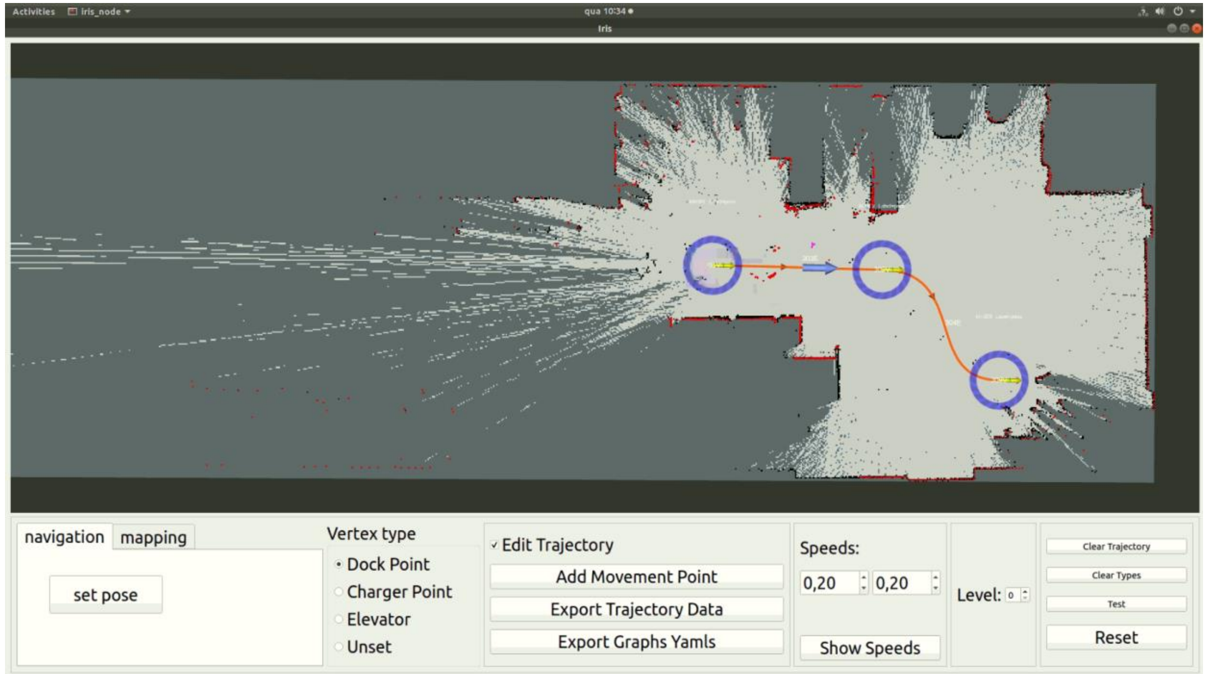
9. If so, it is possible to move the robot by right-clicking on the vertex you want to move the robot to and clicking on the ***Come Here*** button.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation



10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

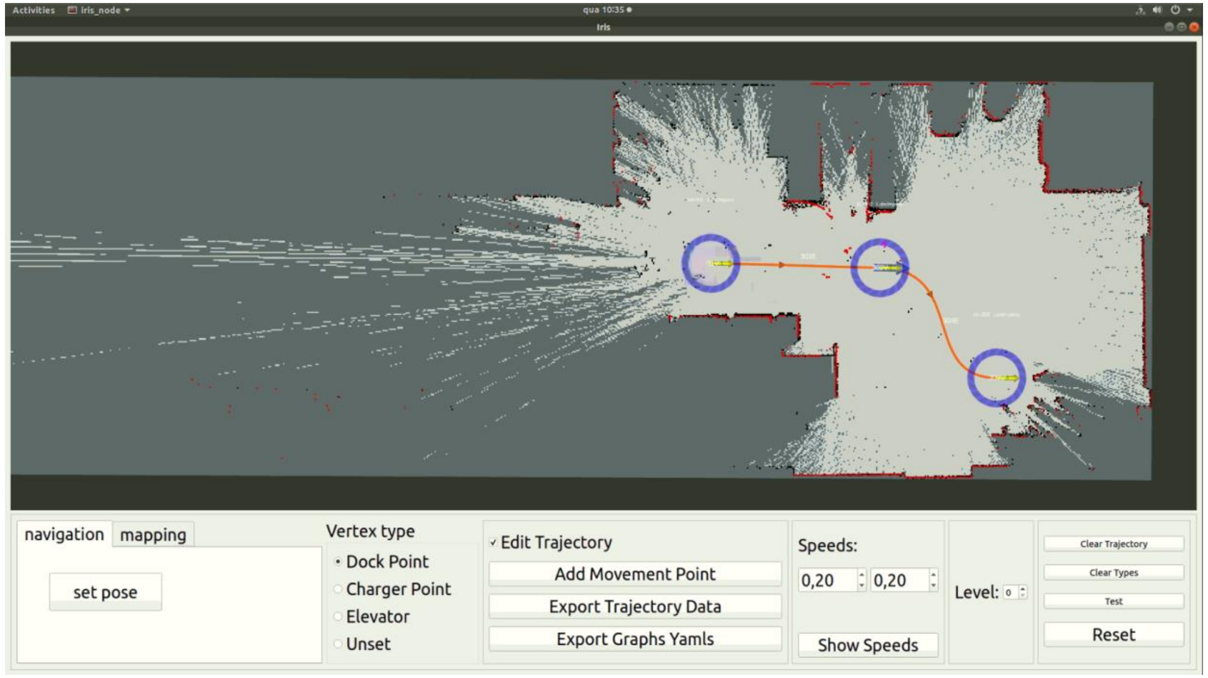
- Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT’S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

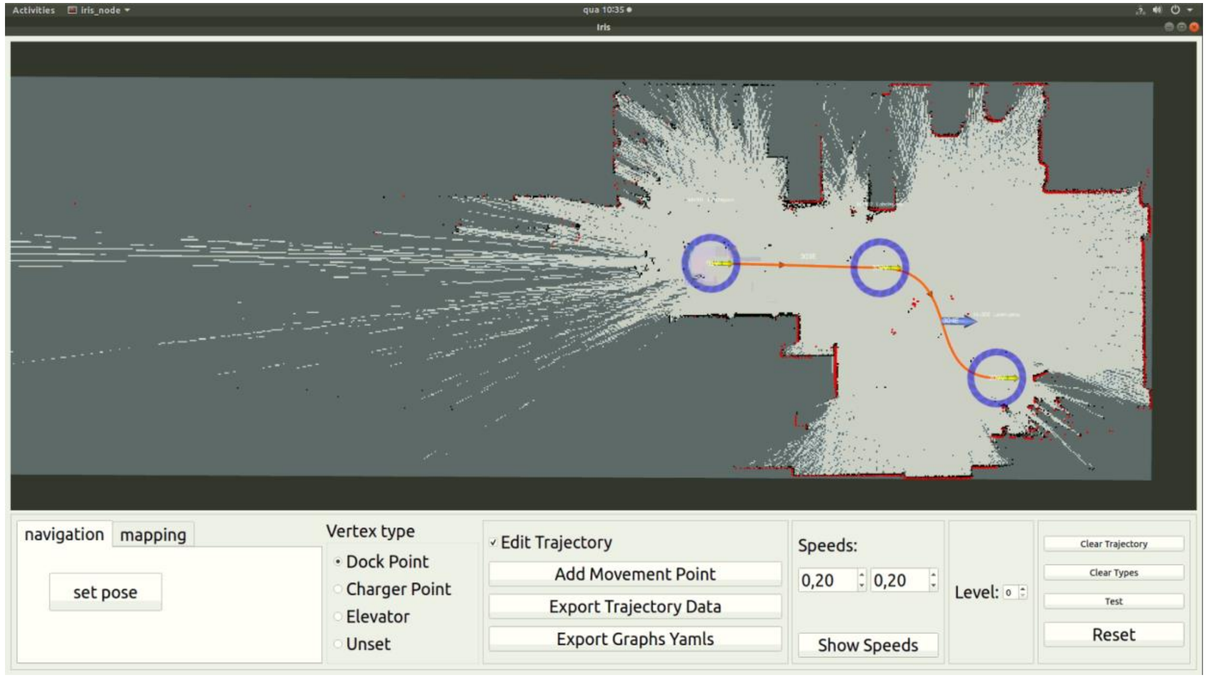
- Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

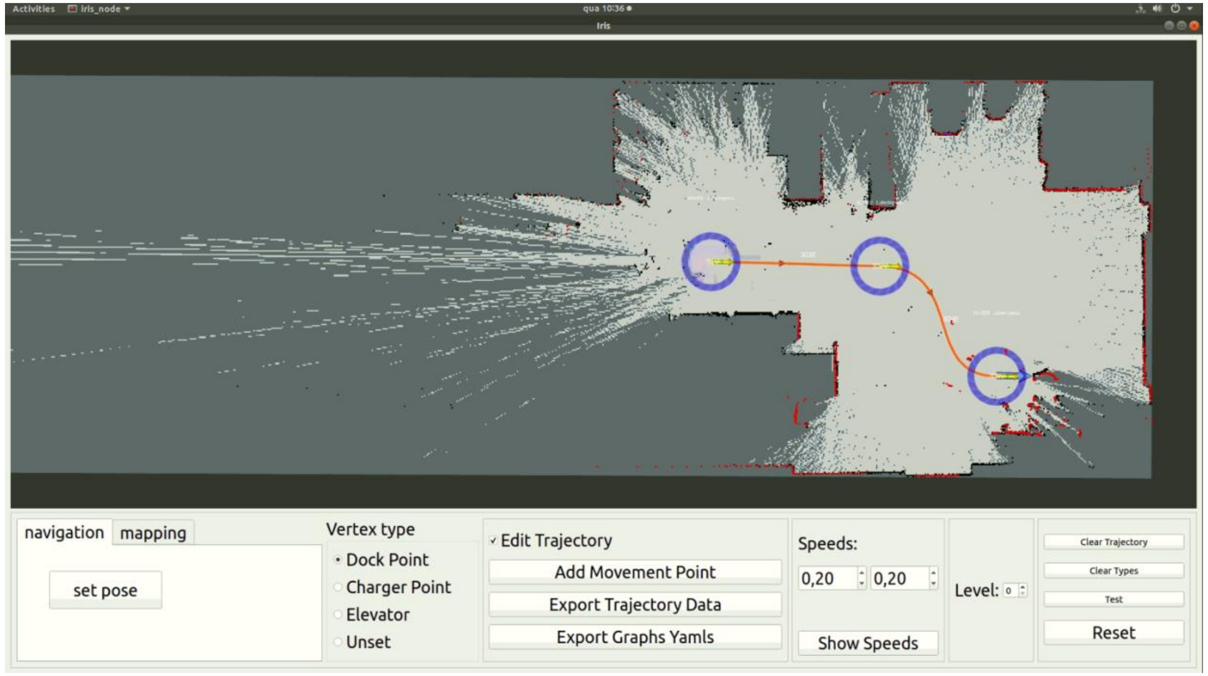
• IRIS Configuration – Navigation Operation

MARI4YARD MARI4ALLIANCE

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Thank you for your attention!

MARI4YARD
MARI4ALLIANCE

Paulo Rebelo | Researcher

paulo.m.rebelo@inesctec.pt

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

Internal Training

Mobile Manipulation for Internal Logistics

June 11 and 12, 2024

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Module 1

Mobile Robot Navigation System Configuration

June 11 and 12, 2024

Paulo Rebelo
Researcher
INESC TEC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Practice User Guide

Navigation Stack User Manual

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

FRIDAY

AUTONOMOUS MOBILE MANIPULATOR

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

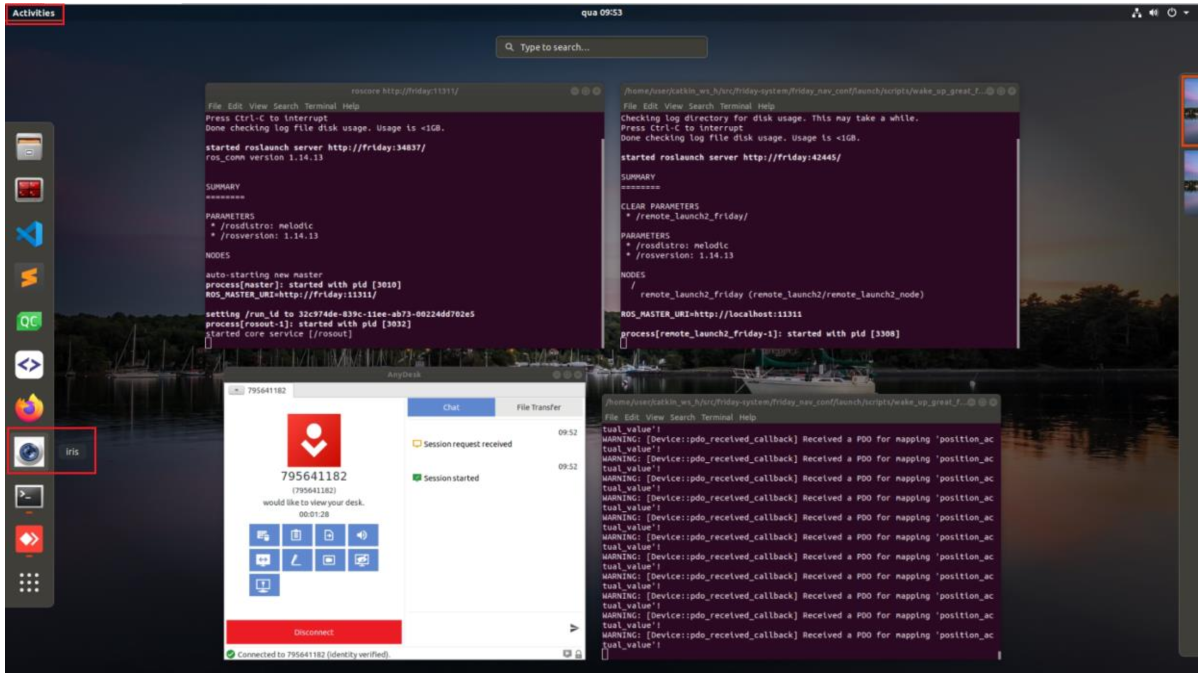
INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration

1. Click on ***Activities*** in the left top screen corner;

2. The left bar will appear. After Click on ***IRIS Application***;



MARI4YARD
MARI4ALLIANCE

- This interface allows the configuration of the navigation system for a new installation or configuration;
- It is where the routes, used by the mobile robots, are mapped and defined.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

The mapping operation is a fundamental part of the system's initial configuration. It consists on the creation of an occupancy matrix that represents the system's operation space.

This matrix delimits the space in which the system can operate and where possible obstacles are located.

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

3. Click on the ***Mapping Tab*** to start, autonomously, the mapping operation;

4. Use the joystick to move the mobile platform in order to visit all the areas in which the system will operate;

During the mapping operation, IRIS displays the generated map in real-time;

5. Once the mapping process is completed, Click on ***Save Map*** button to save the robot's map;

6. If there are more floors to map, move the robot to the new floor, click ***Reset*** and repeat the previous steps.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

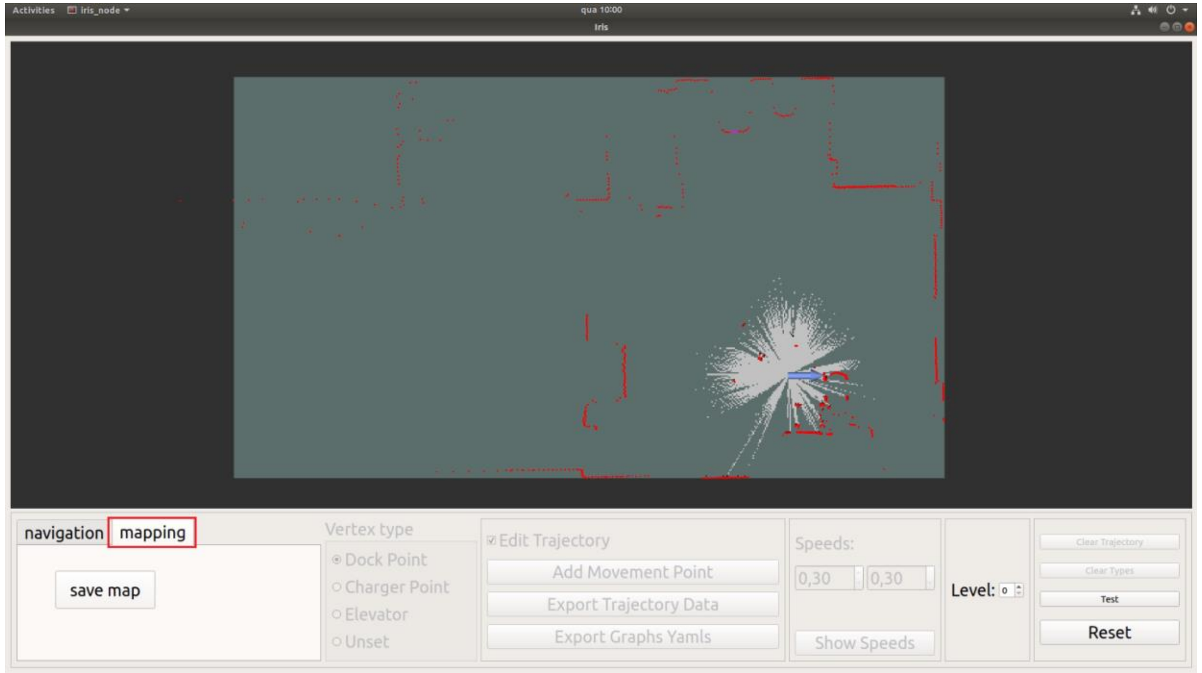
wait until you see something identical to the image on the right on the screen

- Human-Machine Interface Installation and Configuration

- IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

3. Click on the ***Mapping Tab*** to start, autonomously, the mapping operation;



WAIT 2 MINUTES! BE PACIENT!

The system needs time to launch some new packages.

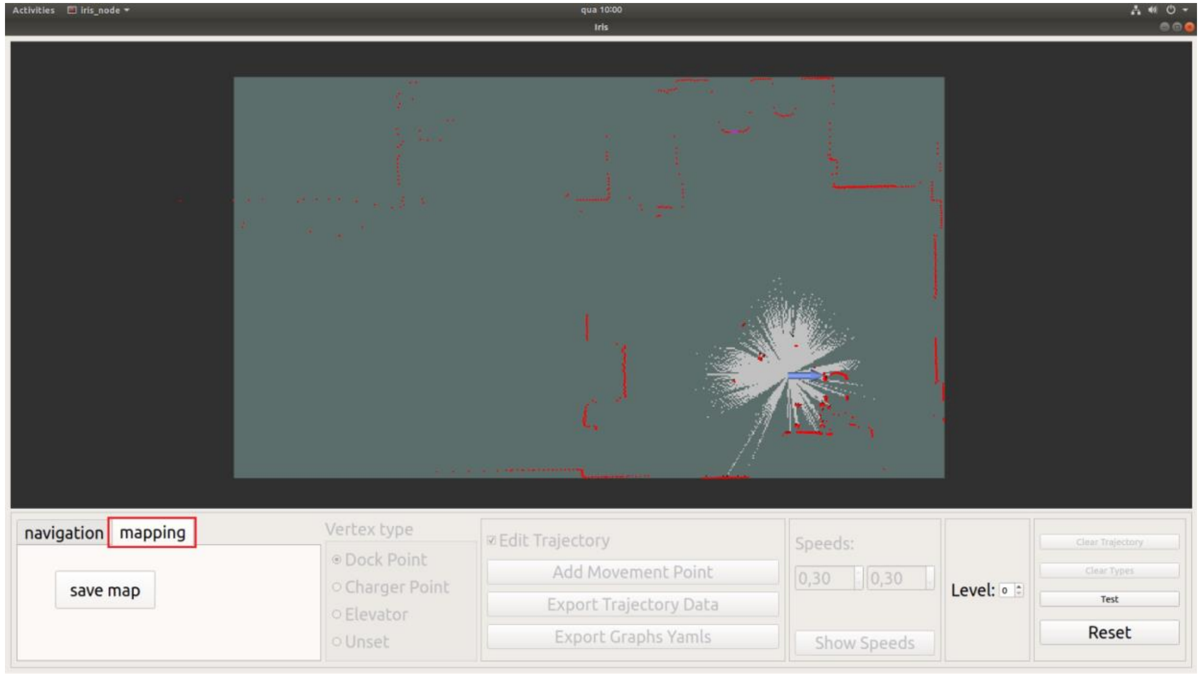
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

wait until you see something identical to the image on the right on the screen

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation



3. After getting some data displayed on the screen make the following steps:

- ***Mouse Scroll*** for Zoom Out or Zoom In;
- ***Hold*** the ***Middle Button*** and drag/center the image.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

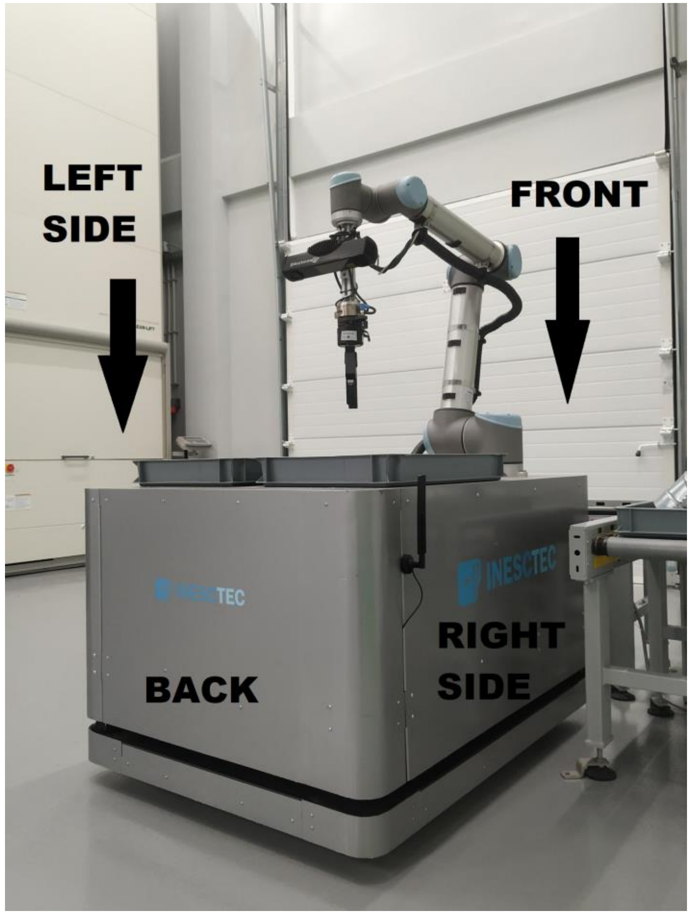
- Human-Machine Interface Installation and Configuration

- IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

PAY ATTENTION TO THE ROBOT'S ORIENTATION!!

Before Move the Robot



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

1. ***Hold Down*** the ***LB*** button to move the robot;

4. Use the joystick to move the mobile platform in order to visit all the areas in which the system will operate.

BE CAREFUL! BE PRUDENT!

During the mapping operation, IRIS displays the generated map in real-time;

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE

2. The ***Left Analogue*** button is used to control the robot forwards, backwards, left and right;

BE CAREFUL! BE PRUDENT!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Mapping Operation

3. The ***Right Analogue*** button is reserved for the rotation movement.

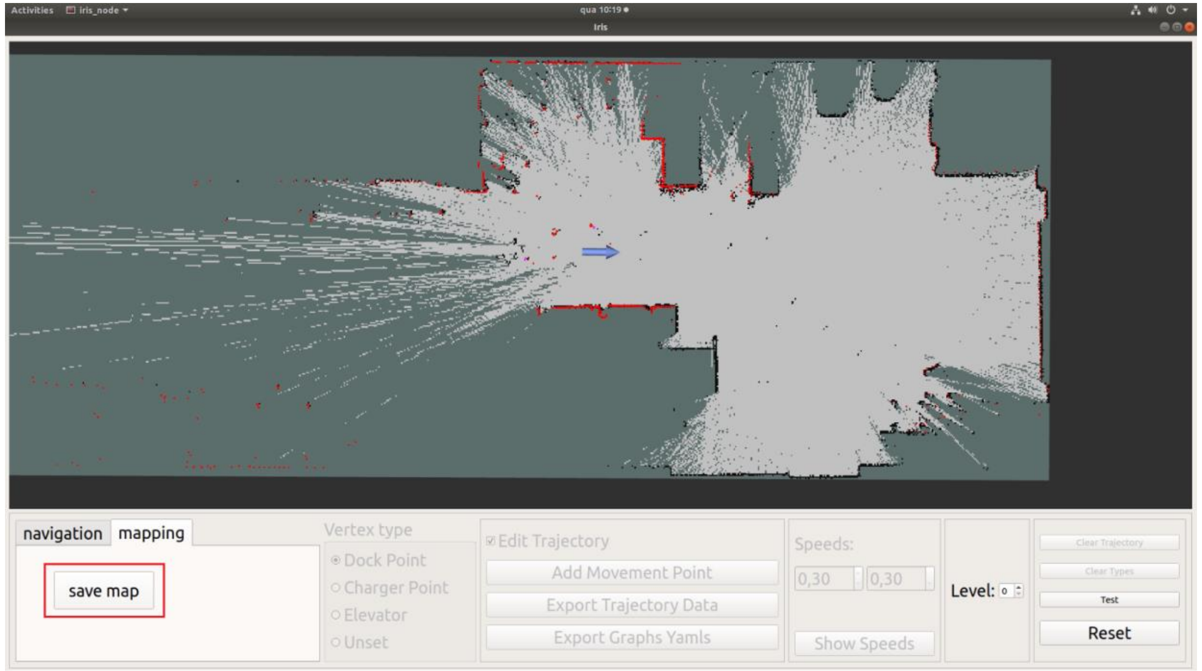
BE CAREFUL! BE PRUDENT!

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Mapping Operation

MARI4YARD
MARI4ALLIANCE



5. Once the mapping process is completed, Click on ***Save Map*** button to save the robot’s map.

WAIT 1 MINUTE! BE PACIENT!

The system takes time to save the new files.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

The navigation operation consists on the creation of a new trajectory for the robot and finally move the robot autonomously.

Edges Types:

- Unidirectional – Robot moves only in one direction;
- Bidirectional – Robot moves in two directions;

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

1. Click on the ***Navigation Tab*** to switch from the previous operation. The system launch, autonomously, the latest created map;

2. Click on the ***Clear Trajectory*** button and then in ***Clear Types*** button to erase the latest stored trajectory. The system create just one vertex in the origin map position;

3. After is necessary locate the mobile platform in the new map. Click on the ***Set Pose*** button and follow the next steps:

   - Move mouse to the supposed robot's position;
   - Hold the Left Mouse Click and then drag it in the orientation the robot is in;
   - If you find that the robot is not located on the new map, repeat the steps again.

4. After located, add the second vertex on robot's pose. Click on the ***Add Movement Point*** button. The system will create a new vertex with the robot's orientation;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

NOTE: These steps will be explained in detail in the following slides. This slide only serves to summarize the actions.

The navigation operation consists on the creation of a new trajectory for the robot and finally move the robot autonomously.

Edges Types:

- Unidirectional – Robot moves only in one direction;
- Bidirectional – Robot moves in two directions;

5. Build the path by creating and configuring new vertices;

6. Edit the path by creating and configuring the edges between vertices;

7. Once the path configuration process is completed, Click on ***Export Trajectory Data*** button and on the ***Export Graph Yamls*** button to save the robot's trajectory;

8. After these steps, click on the ***RESET*** button and check that the robot is located and that the new map and trajectory are launched autonomously and correctly.

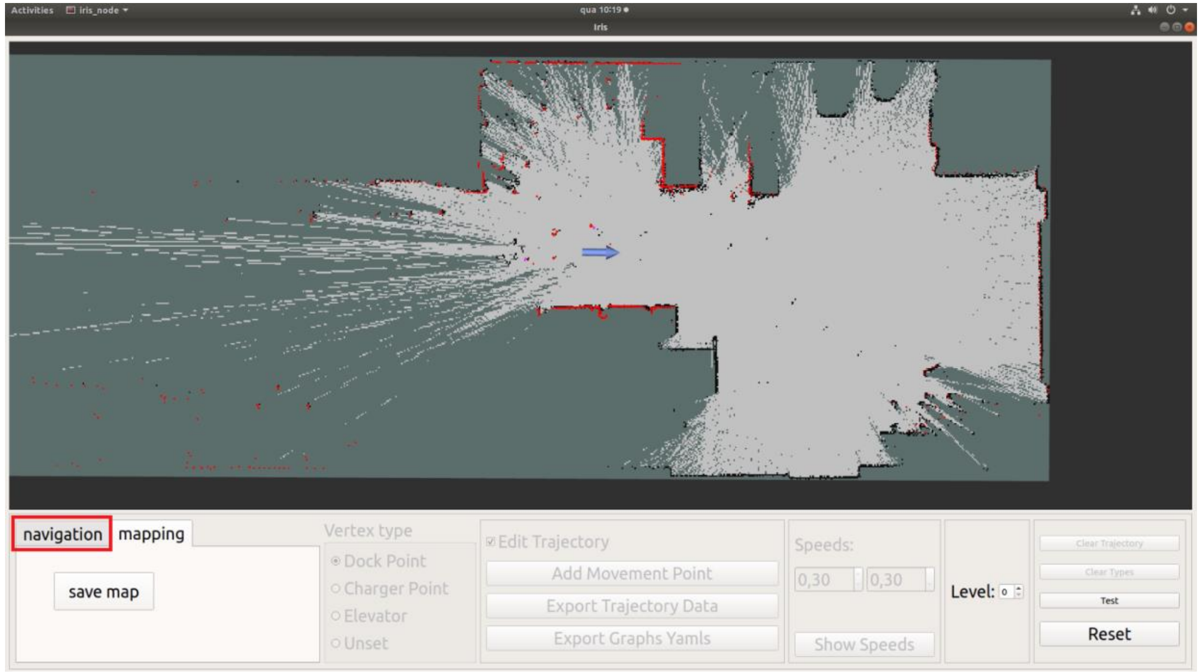
9. If so, it is possible to move the robot by right-clicking on the vertex you want to move the robot to and clicking on the ***Come Here*** button.

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation



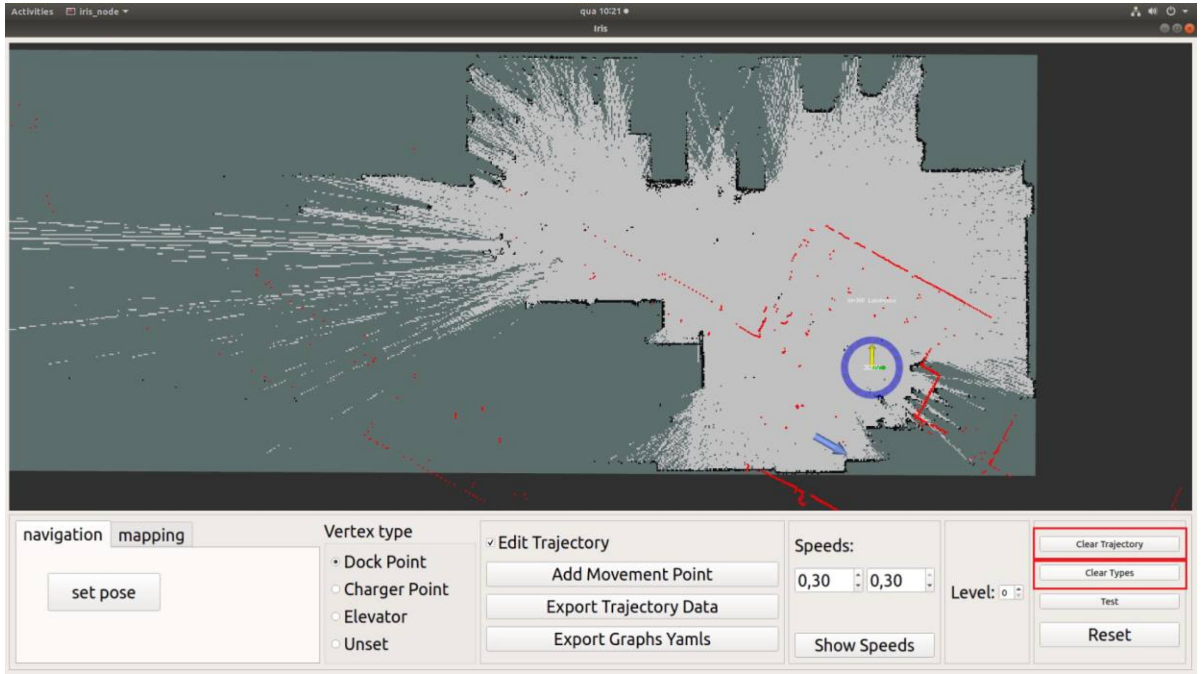
1. Click on the ***Navigation Tab*** to switch from the previous operation. The system launch, autonomously, the latest created map;

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation

2. Click on the ***Clear Trajectory*** button and then in ***Clear Types*** button to erase the latest stored trajectory. The system create just one vertex in the origin map position;



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

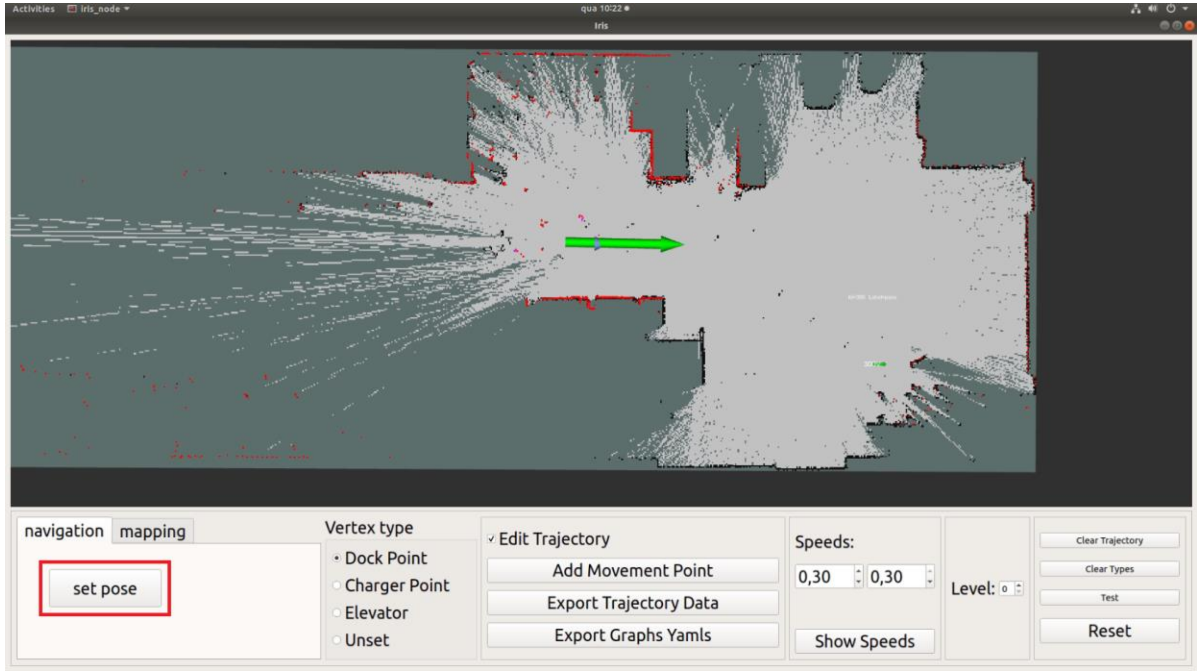
• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

3. After is necessary locate the mobile platform in the new map. Click on the ***Set Pose*** button and follow the next steps:

- Move mouse to the supposed robot's position;
- ***Hold*** the ***Left Mouse*** Click and then drag it in the orientation the robot is in;
- If you find that the robot is not located on the new map, repeat the steps again.

NOTE: The robot is located only when the red lasers points match with the black wall points on the new map.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

4. After located, add the second vertex on robot's pose. Click on the ***Add Movement Point*** button. The system will create a new vertex with the robot's orientation;

NOTE: Inside the green rectangle is possible to see the new vertex and the robot located. The red points are the real-time lasers points data.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

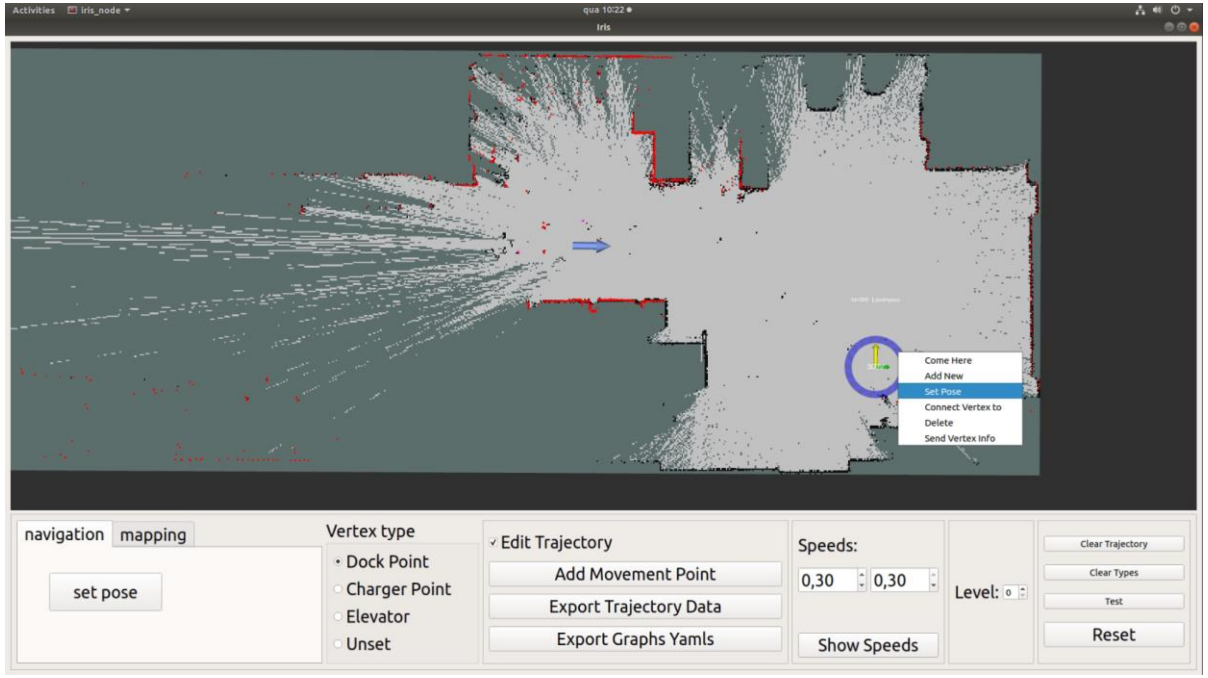
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

5. Build the path by creating and configuring new vertices, following the next steps:

- ***Right Mouse Click*** on the blue circle of a vertex;
- Click on ***Add New***;
- ***Hold*** the ***Left Mouse Click*** on the green arrow, on the created vertex, and move it to the desired map point;
- ***Hold*** the ***Left Mouse Click*** on the blue circle of the moved vertex and rotate the circle until you get the direction of the desired trajectory, represented by the green arrow direction;
- Finally, adjust the yellow arrow alluding to the robot's orientation at that waypoint/vertex by ***Holding*** the ***Left Mouse Click*** on the yellow arrow and rotating it until the desired orientation is obtained.



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

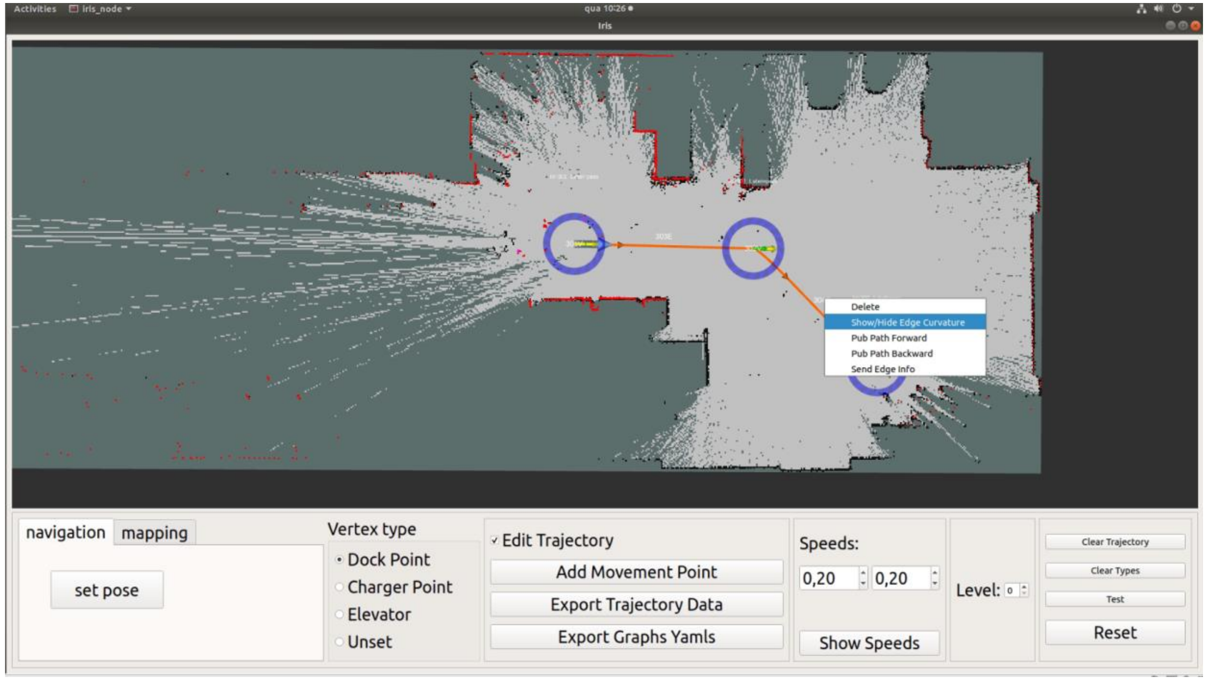
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

6. Edit the path by creating and configuring the edges between vertices, following the next steps:

- ***Right Mouse Click*** on the blue circle of a vertex;
- Click on ***Connect Vertex To***;
- ***Left Mouse Click*** on the blue circle of the second vertex - it is essential that the two vertices do not have opposite orientations;
- The new Edge is created between the two defined vertices;
- ***Right Mouse Click*** on the orange line of an edge to adjust it;
- Click on ***Show/Hide Edge Corvature***. (**See Next Slide)**



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

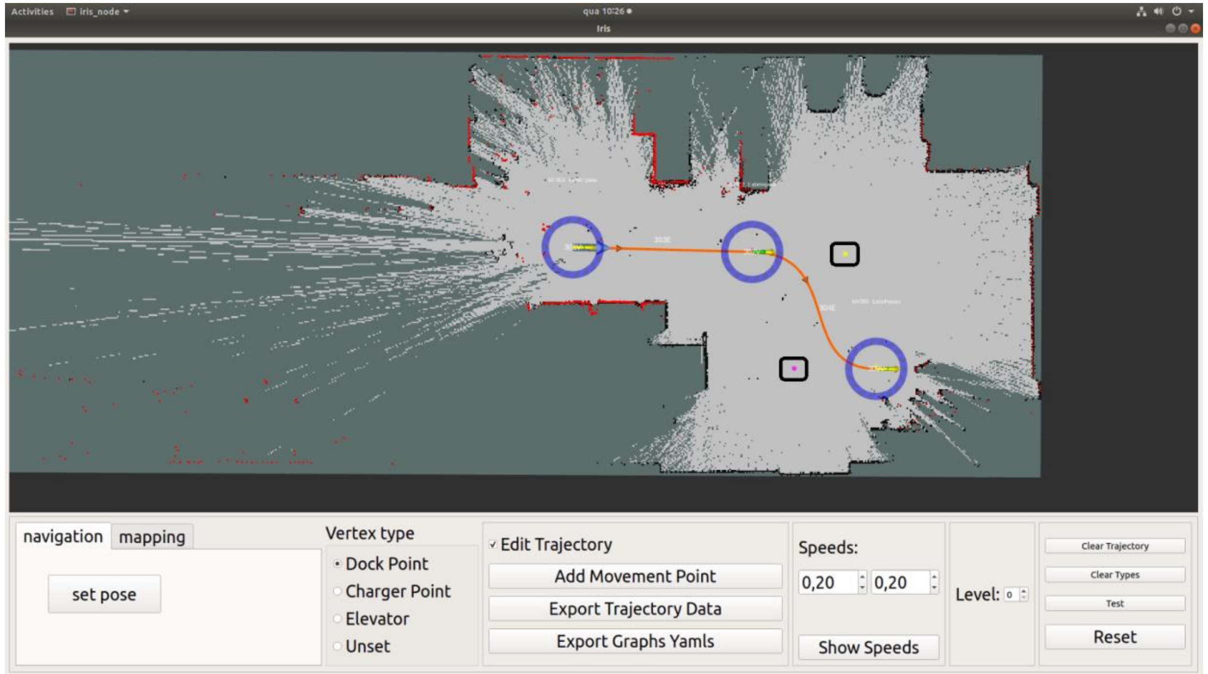
• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

6. Edit the path by creating and configuring the edges between vertices, following the next steps:

- ***Holding*** the ***Left Mouse Button*** adjust, sliding, the yellow and the pink dots - the aim is to make the curve as smooth as possible;
- ***Right Mouse Click*** on the same orange line of the respective edge;
- Click on ***Show/Hide Edge Corvature*** for the dots to disappear from the image.



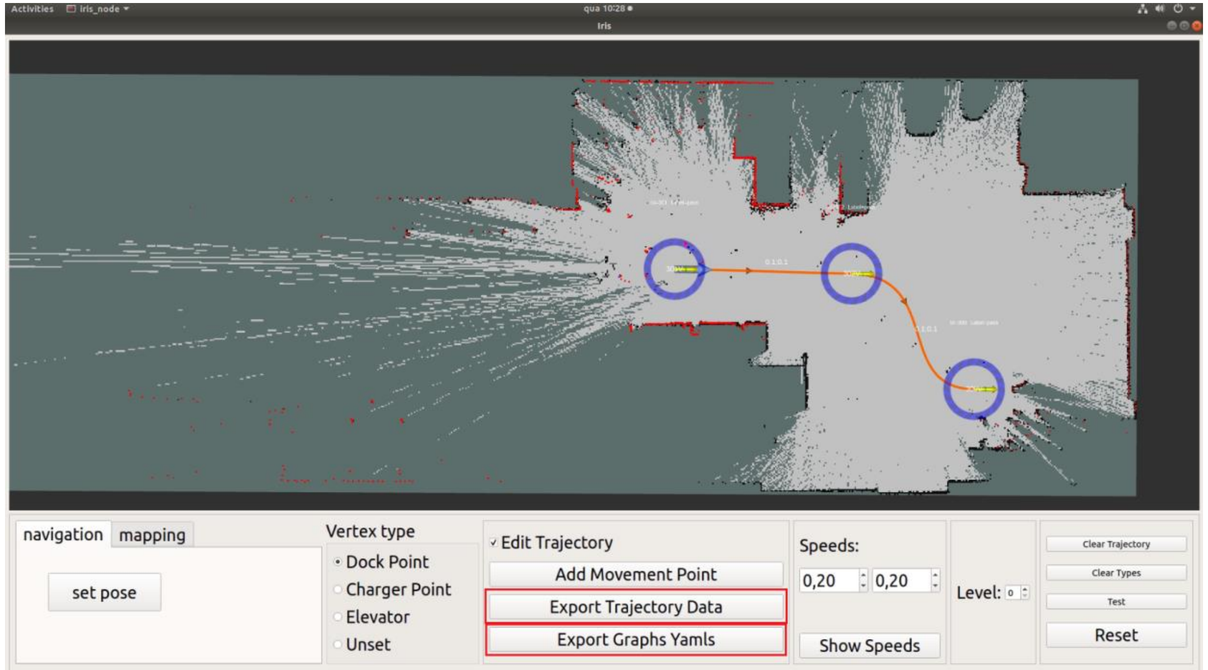
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

7. Once the path configuration process is completed, Click on ***Export Trajectory Data*** button and on the ***Export Graph Yamls*** button to save the robot's trajectory;



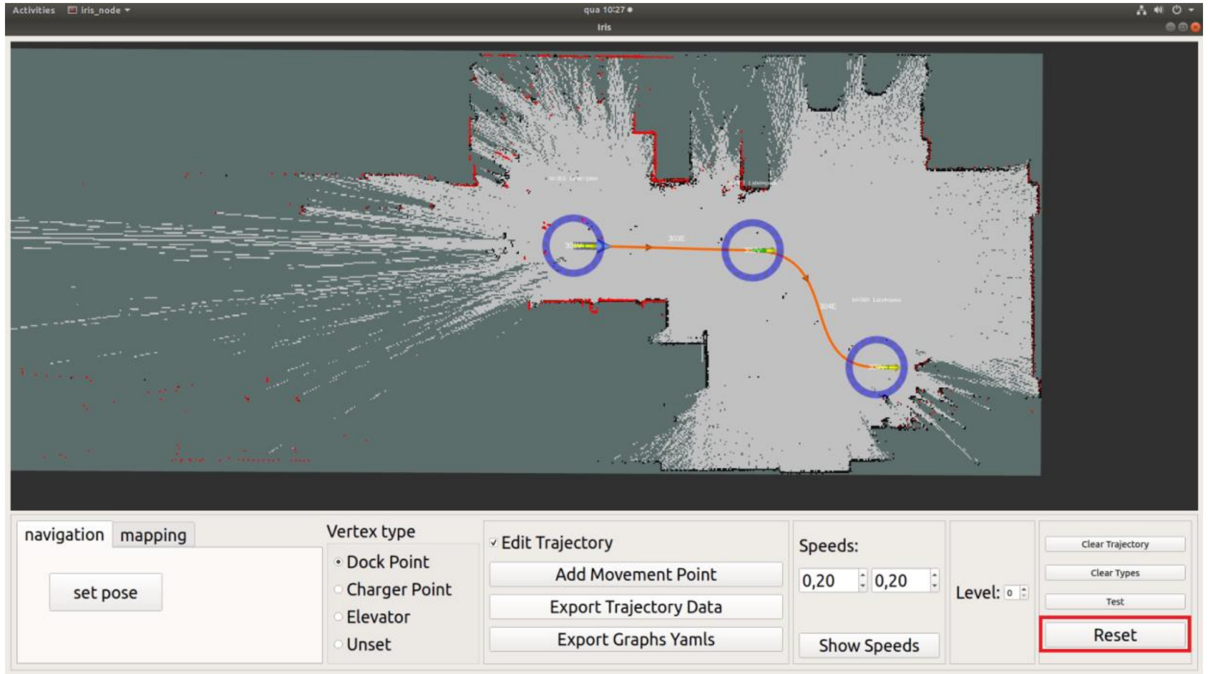
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

8. After these steps, click on the ***RESET*** button and check that the robot is located and that the new map and trajectory are launched autonomously and correctly.



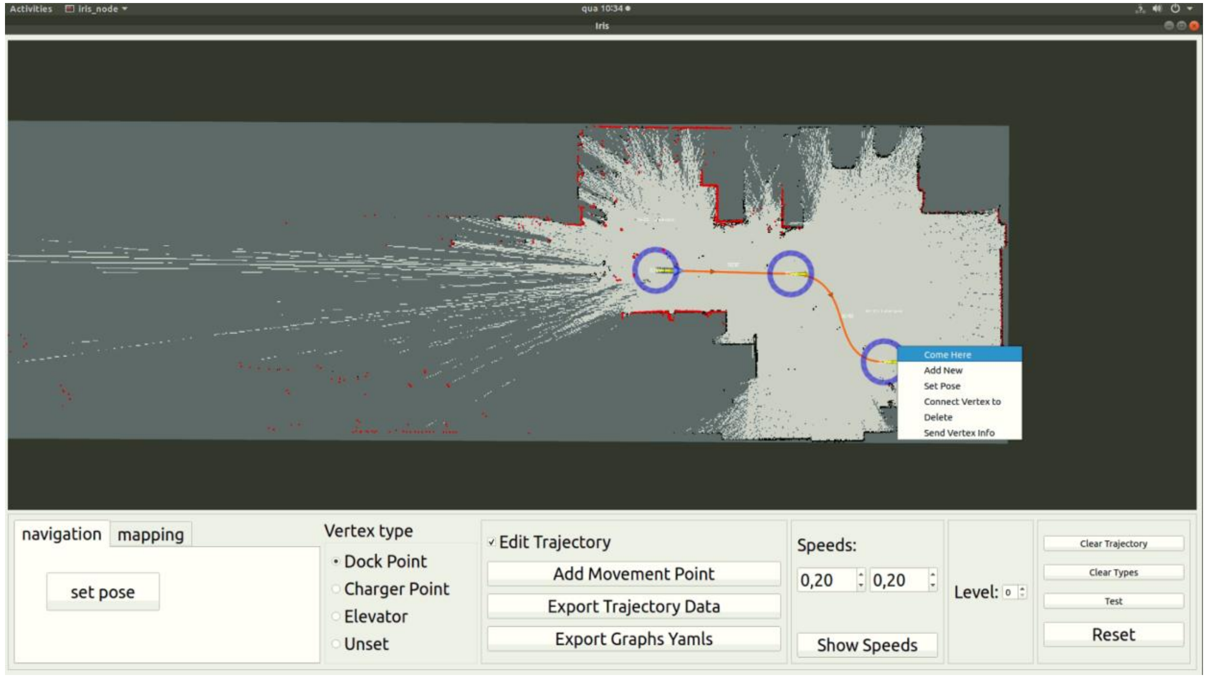
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

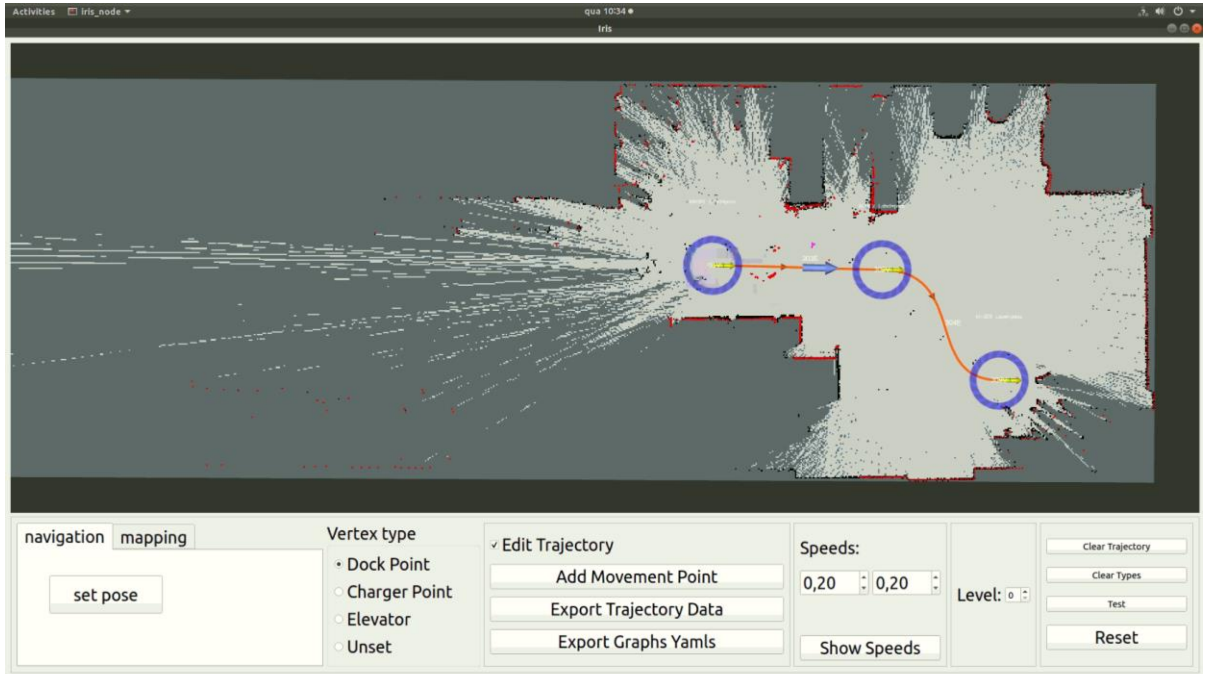
9. If so, it is possible to move the robot by right-clicking on the vertex you want to move the robot to and clicking on the ***Come Here*** button.



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

- IRIS Configuration – Navigation Operation



10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

• Human-Machine Interface Installation and Configuration

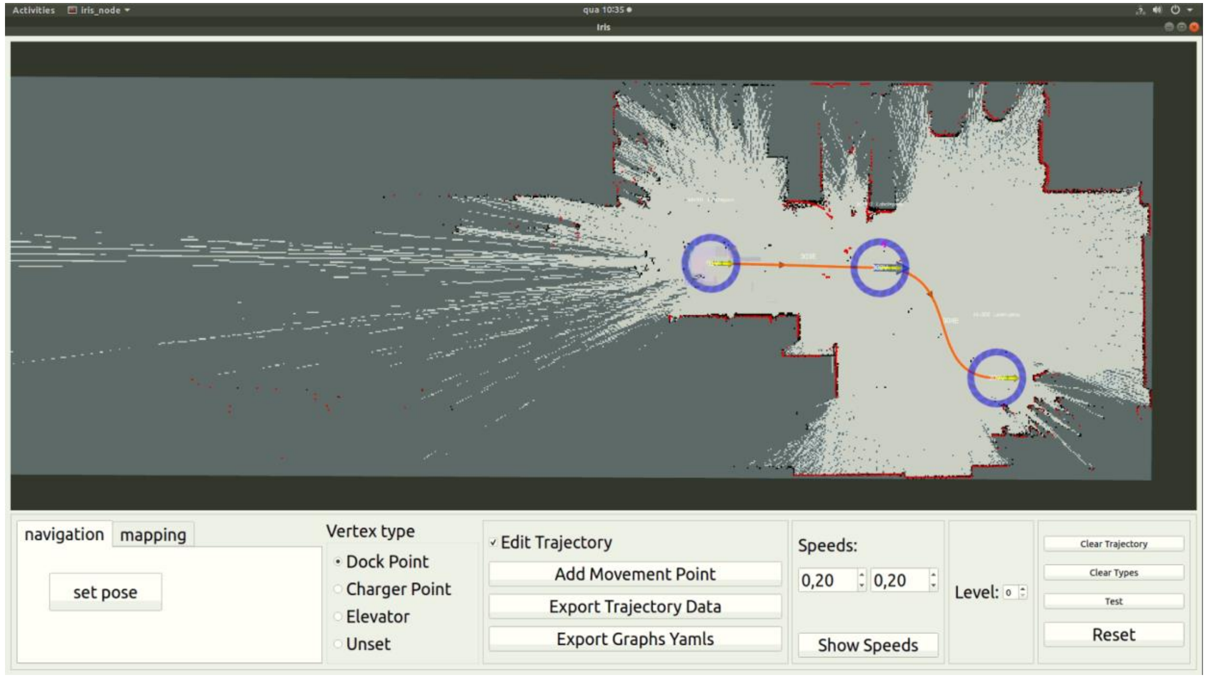
• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE

10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!



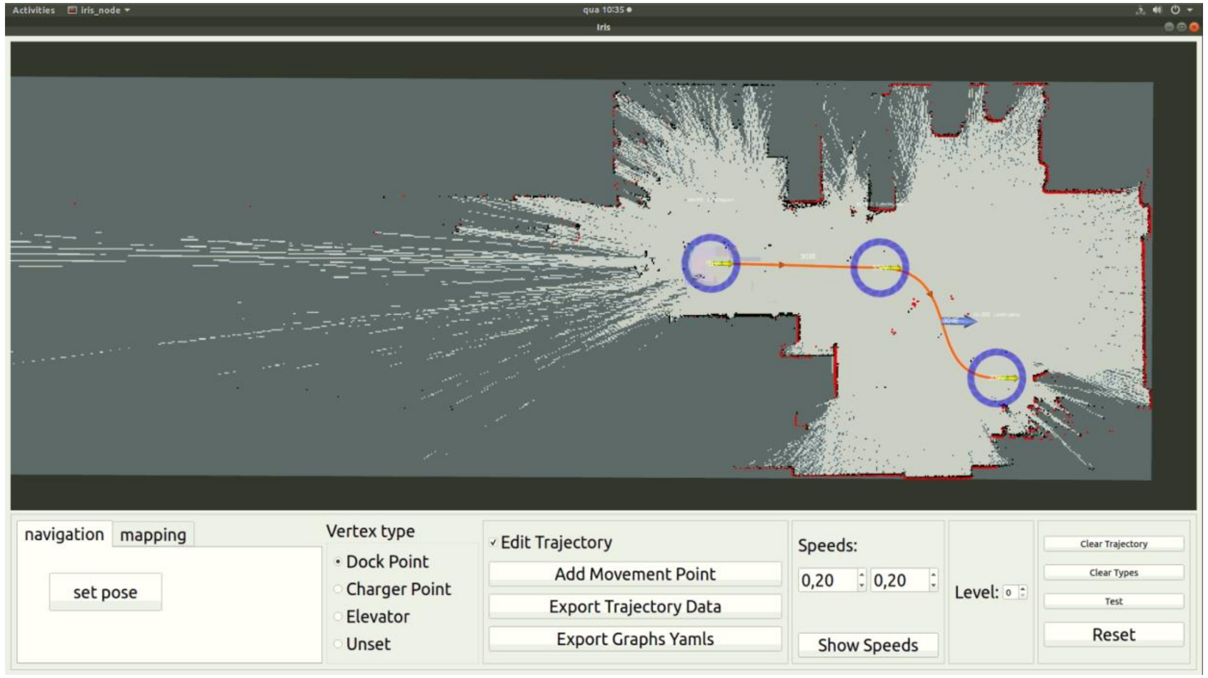
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation

MARI4YARD
MARI4ALLIANCE



10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

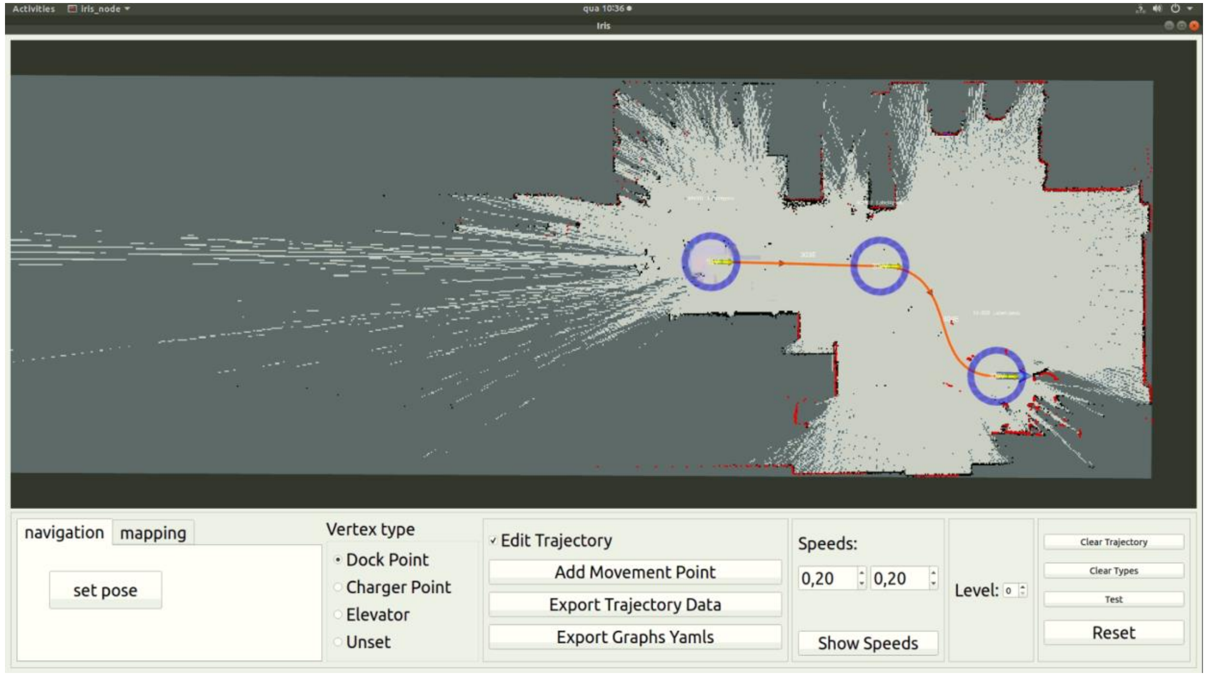
IF NECESSARY PRESS THE EMERGENCY BUTTON!!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

INESCTEC

• Human-Machine Interface Installation and Configuration

• IRIS Configuration – Navigation Operation



10. The robot moves from where it is to the desired point autonomously if there is a path for it to get there.

PAY ATTENTION TO THE ROBOT'S MOVEMENT!

IF NECESSARY PRESS THE EMERGENCY BUTTON!!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Thank you for your attention!

Paulo Rebelo | Researcher

paulo.m.rebelo@inesctec.pt

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Workshop OSPS

Pedro Melo pedro.m.melo@inesctec.pt

Rafael Arrais rafael.l.arrais@inesctec.pt

Sérgio Marinho sergio.d.marinho@inesctec.pt

INESC TEC – Porto | CRIIS

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Workshop Summary

- Open Scalable Production System
- OSPS Messages
- Skill Generator
- Task Manager – Configuration and Launching
- Production Manager:
  - Connection and Monitoring
  - Task Creator
  - Task Execution

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Prerequisites (not critical)

- You are familiar with ROS concepts such as *messages*, *topics*, pub-*sub*, *services* and *actions*;
- You know how to setup and source ROS workspaces;
- You know how to develop, compile, install, and launch ROS applications;
- You are familiar with the Python programming language and basic OOP concepts;
- You are familiar with Docker.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Open Scalable Production System

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Principles

Skill-based Programming

Reduce costs inherent to adapting robotic applications.

Task Orchestration

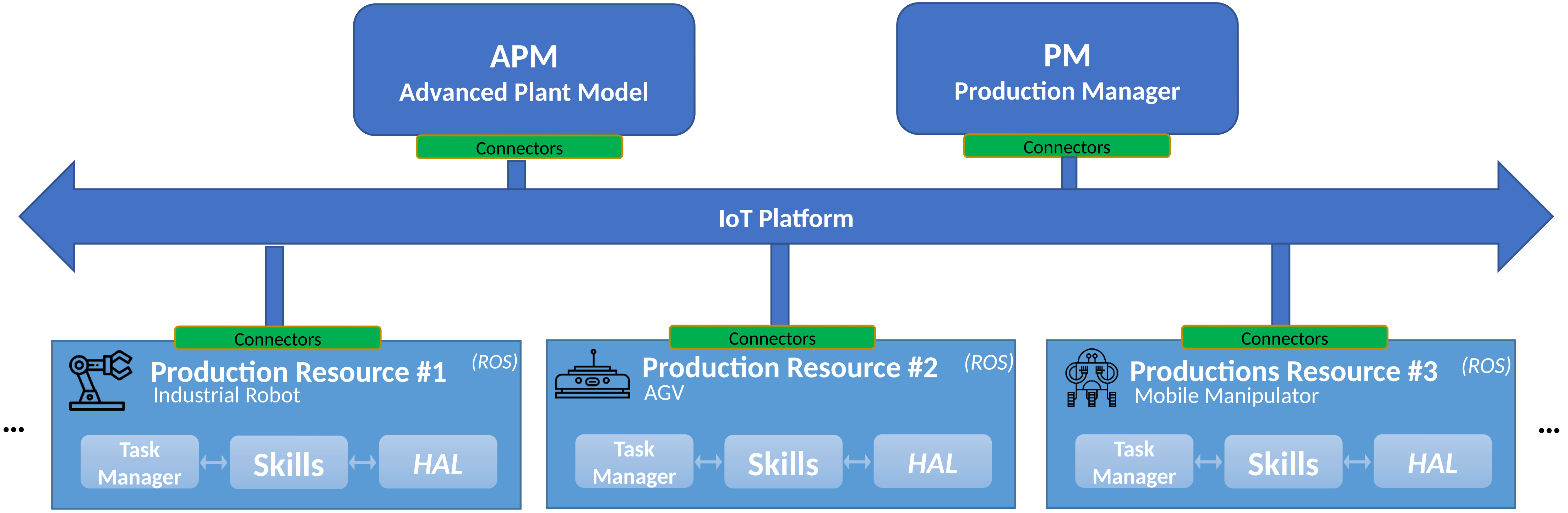
Programming robotic applications in a very intuitive and flexible way.

Vertical and Horizontal Integration

Interoperability with Manufacturing Management Systems and industrial equipment.

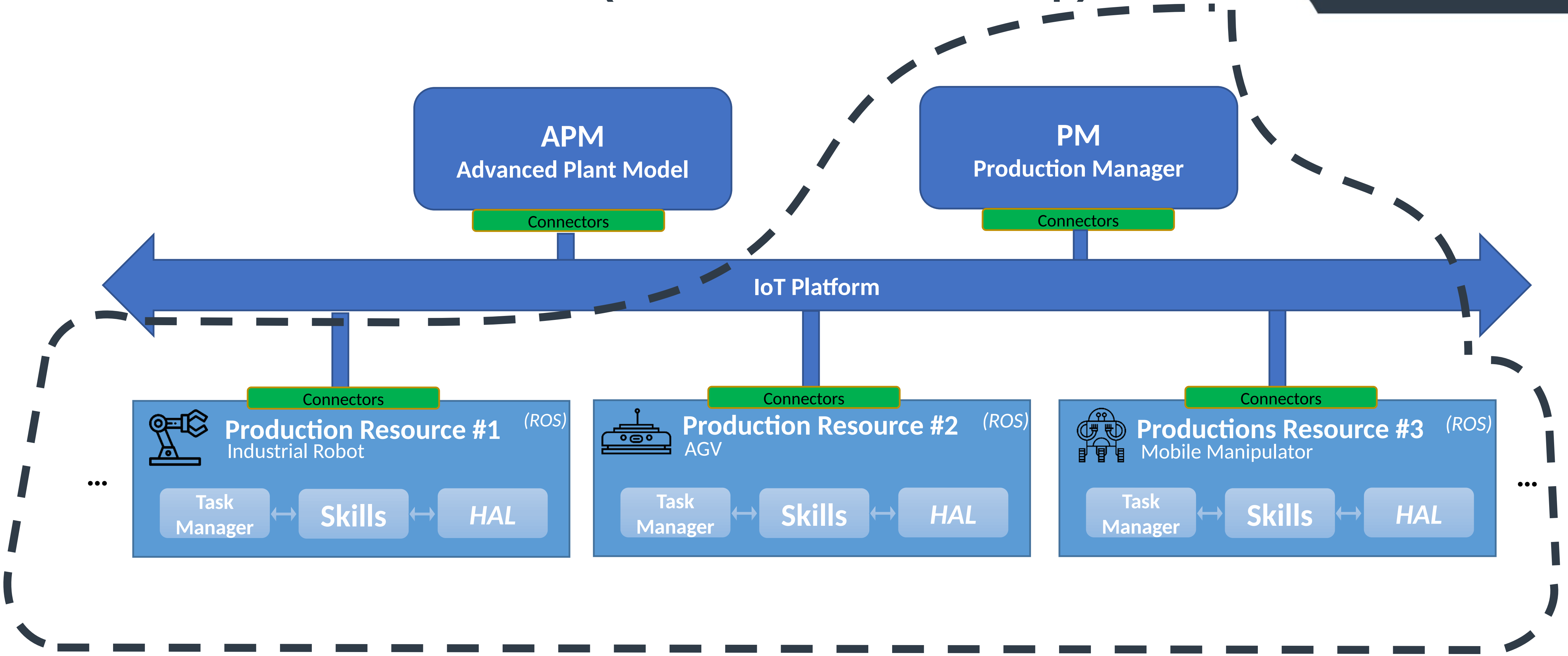
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Overall Architecture



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Overall Architecture (focus of this workshop)

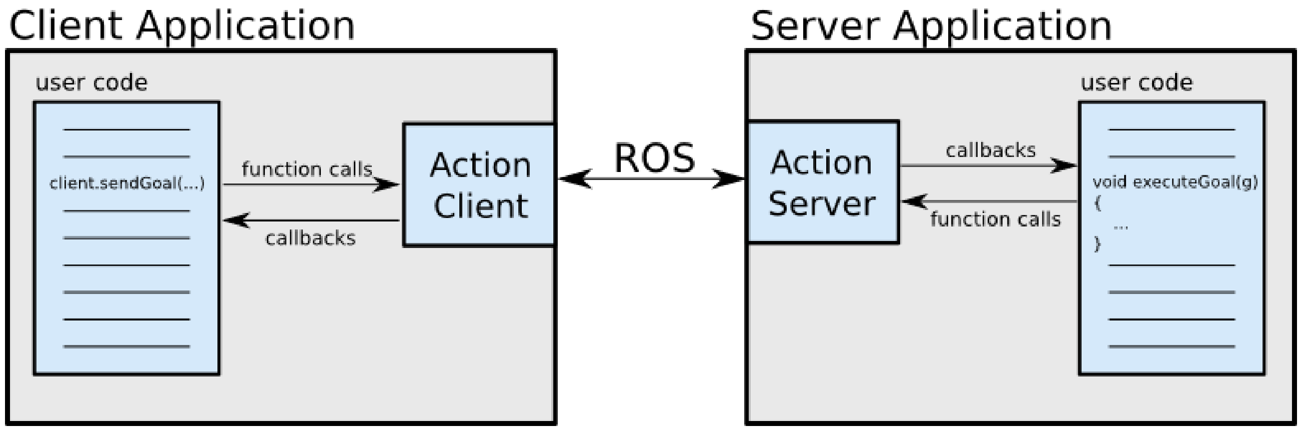


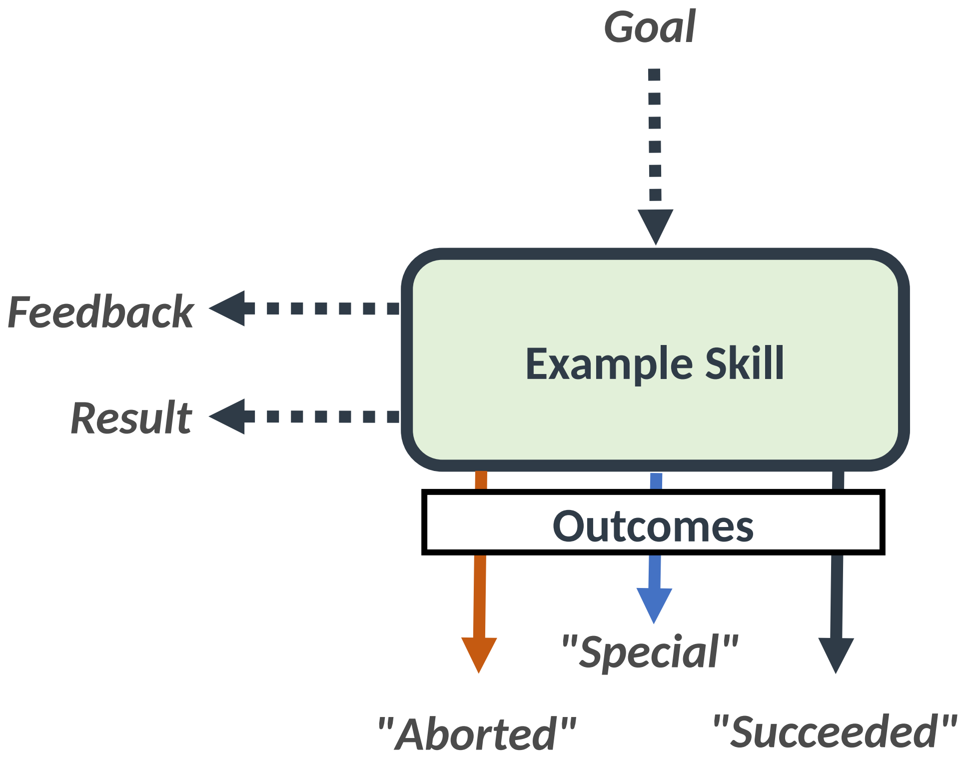
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Skills

Each Skill is an individual <u>ROS package</u> responsible for the execution of a <u>*single unit of basic behavior*</u>

All Skills are built upon *ROS actions – Client* and *Server* components



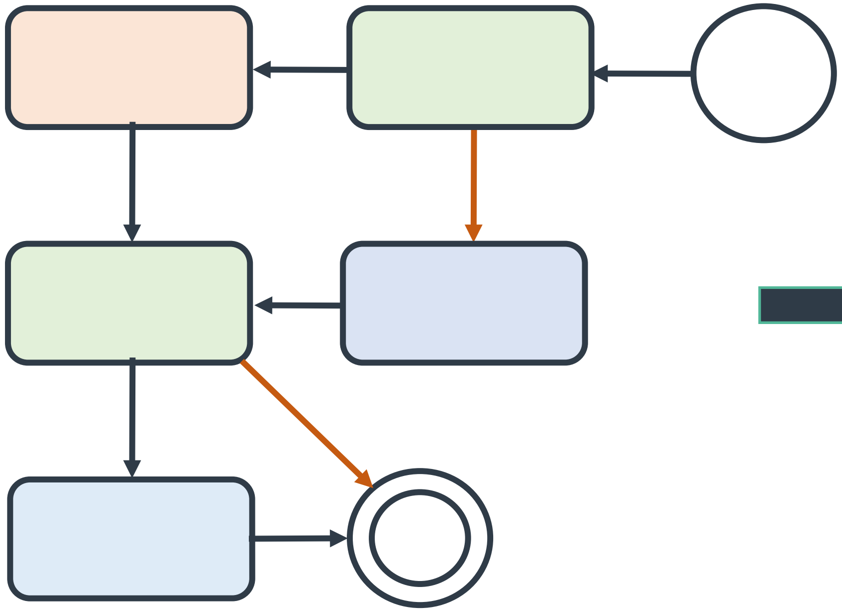


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Tasks

Each Task is a state machines where:

- States are individual Skills with an associated goal;
- Transitions are possible Skill outcomes.



All Tasks are stored as *SCXML* (State Chart XML) files

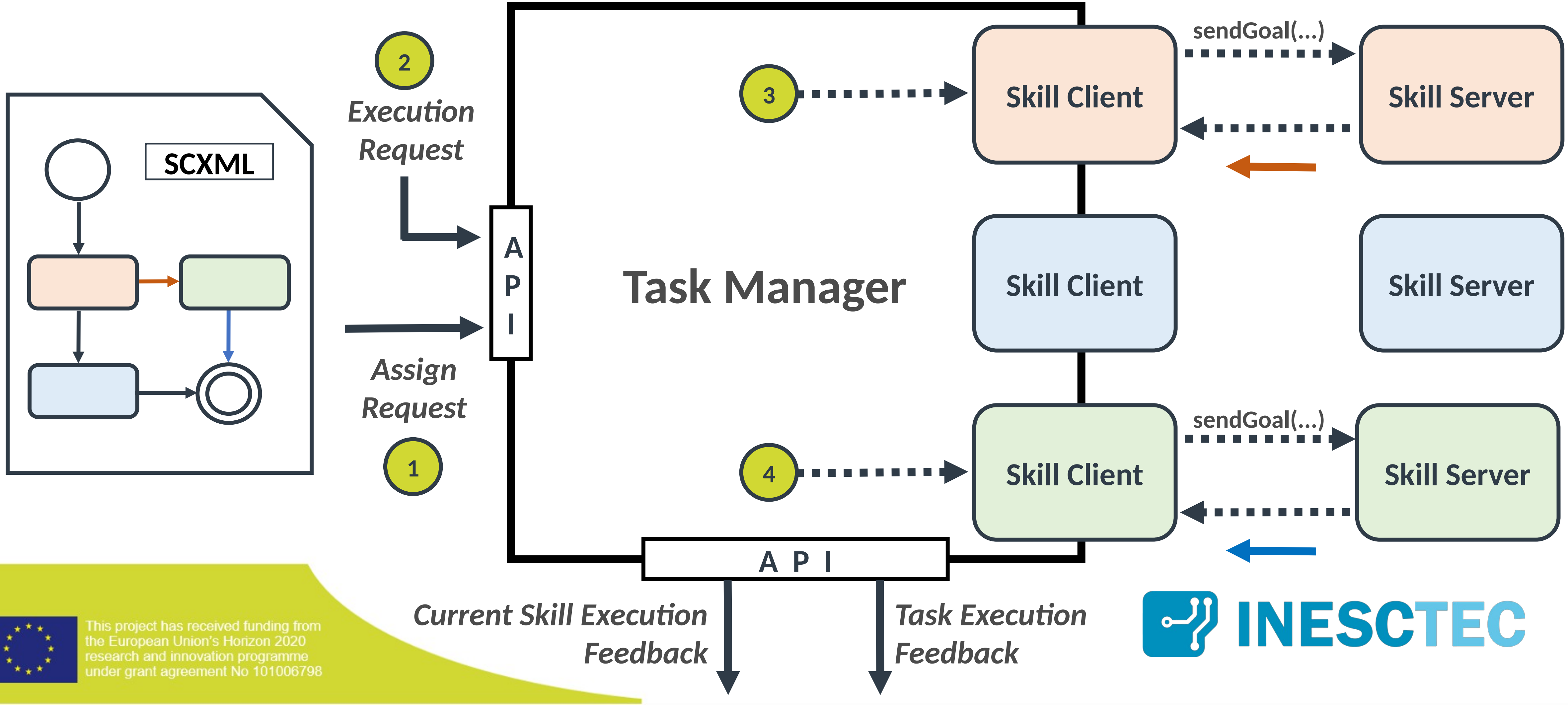
```
<?xml version="1.0" encoding="UTF-8"?>
<scxml initial="WaitSkill">
  <initial_state id_="3" initial_statekey="identifier" id="Initial">
    <Extreme style="initial" parent="1">
      <mxGeometry x="90" y="100" width="40" height="40" as="geometry"/>
    </Extreme>
    <transition id_="10" event="" target="WaitSkill" source="Initial" cond="" id="external">
      <Transition parent="1" source="3" target="8">
        <mxGeometry relative="1" as="geometry"/>
      </Transition>
    </transition>
  </initial_state>
  <state id_="2" id="ExampleSkill">
    <datamodel>
      <data id="actionName" expr="ExampleSkill"/>
      <data id="actionGoal" expr="{&quot;field&quot;:&quot;example&quot;}"/>
      <data id="actionType" expr="example_skill_msgs/ExampleSkillAction"/>
      <data id="actionResult" expr="{}"/>
      <data id="outcomes" expr="[]"/>
    </datamodel>
    <Skill style="skill" parent="1">
      <mxGeometry x="360" y="90" width="80" height="40" as="geometry"/>
    </Skill>
    <transition id_="13" event="succeeded" target="ExampleSkill_3" source="ExampleSkill" cond="" id="external">
      <Transition parent="1" source="2" target="5">
        <mxGeometry relative="1" as="geometry">
          <Array as="points">
            <mxPoint x="400" y="170"/>
            <mxPoint x="236" y="170"/>
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Task Manager

A ROS package that receives SCXMLs, orchestrates and monitors the execution of Manufacturing Tasks

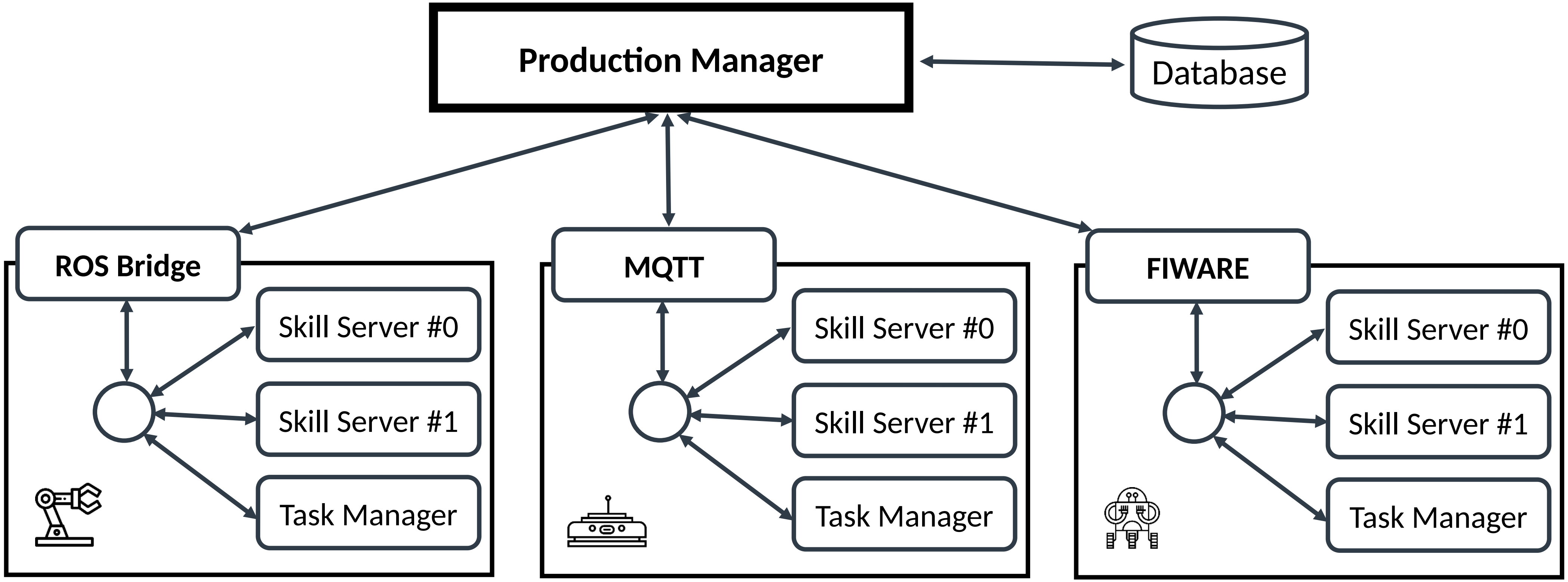
Each robot must run it's own instance of the Task Manager



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS – Production Manager

An application that simplifies Task creation and interaction with multiple Task Manager instances.

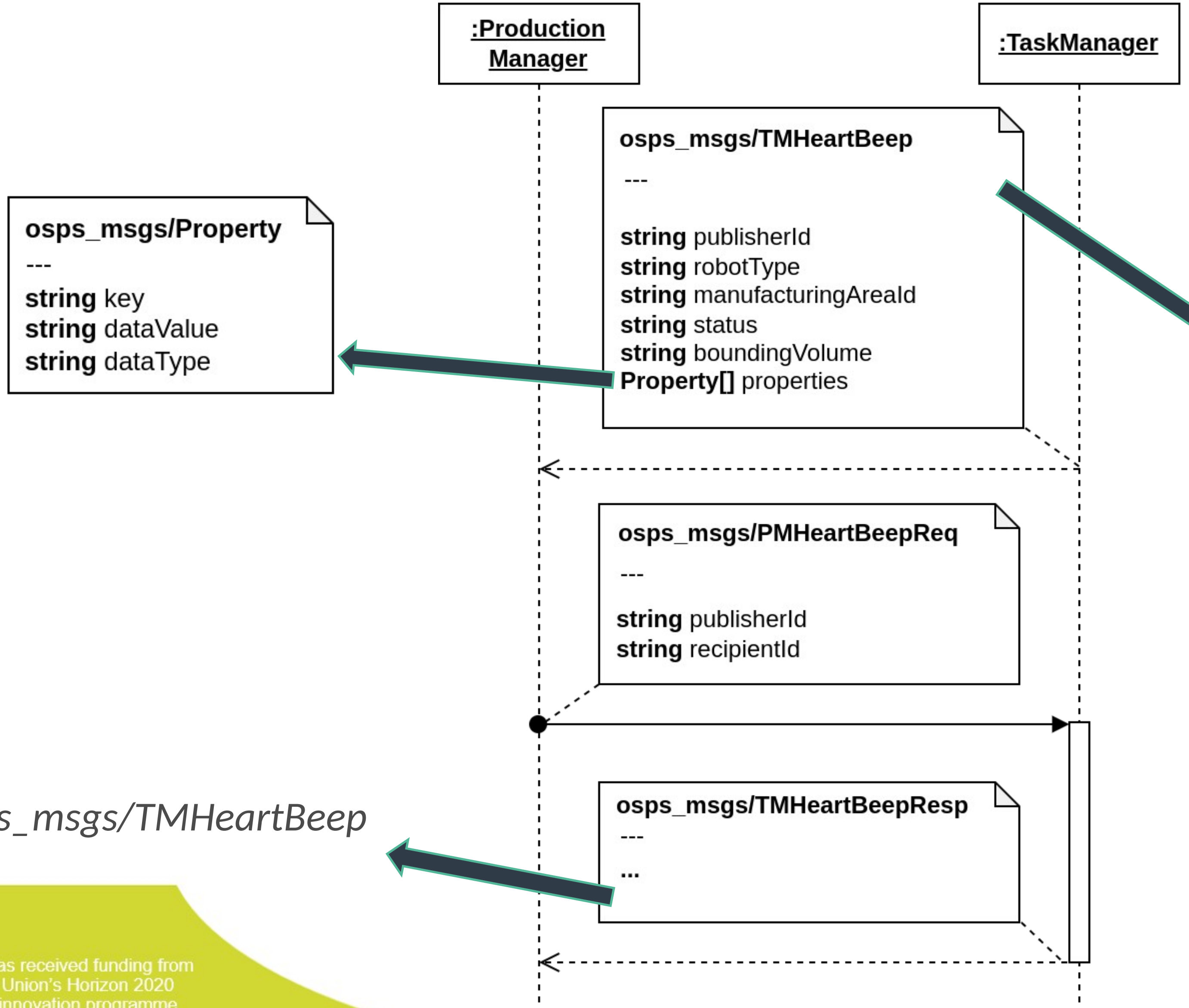


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS Messages

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS Messages – Robot health check



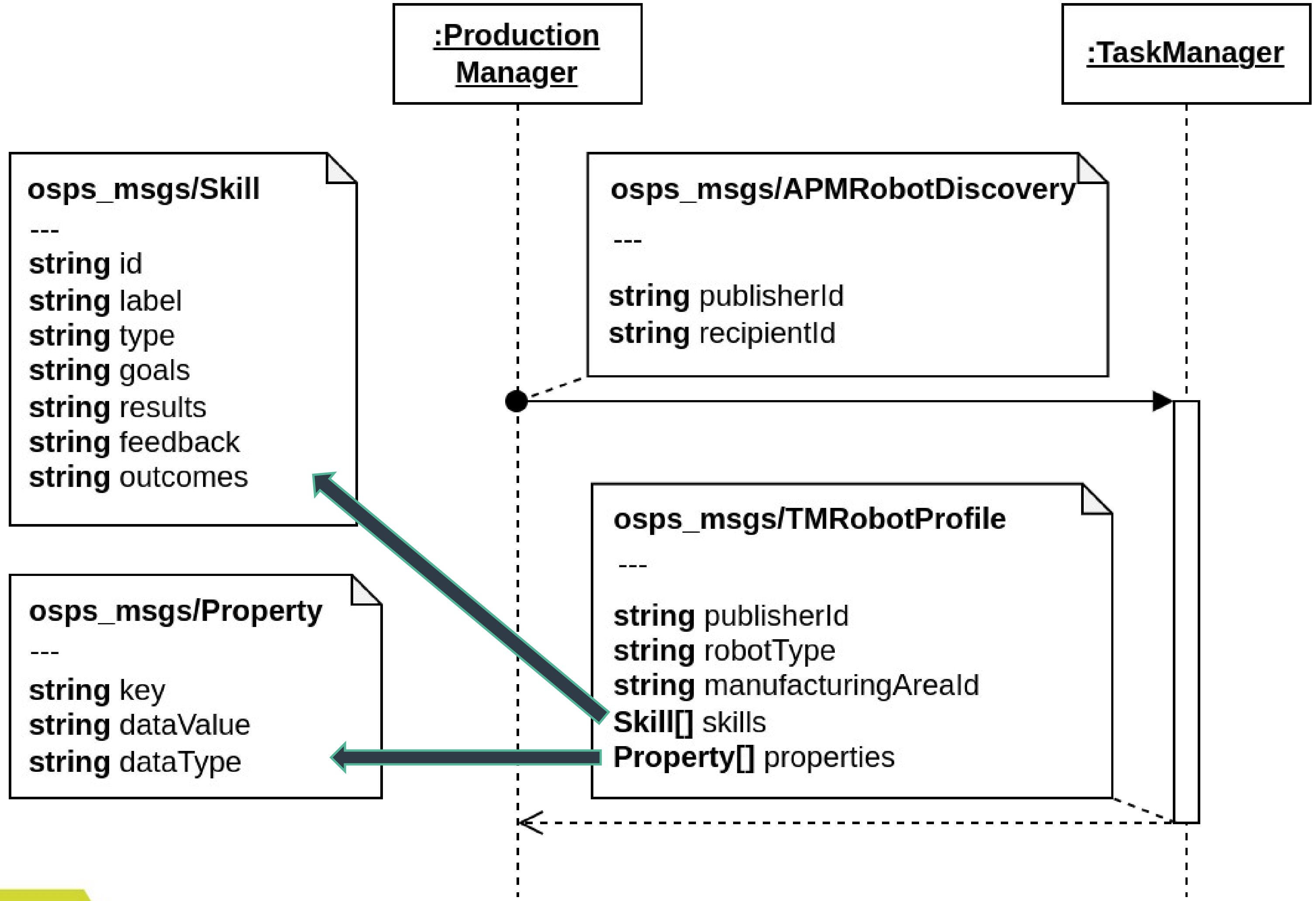
osps_msgs/TMHeartBeep

Field *publisherId* **must** be set manually. All other robot information may also be customized.

Robot location *(boundingVolume)* is given based on TF.

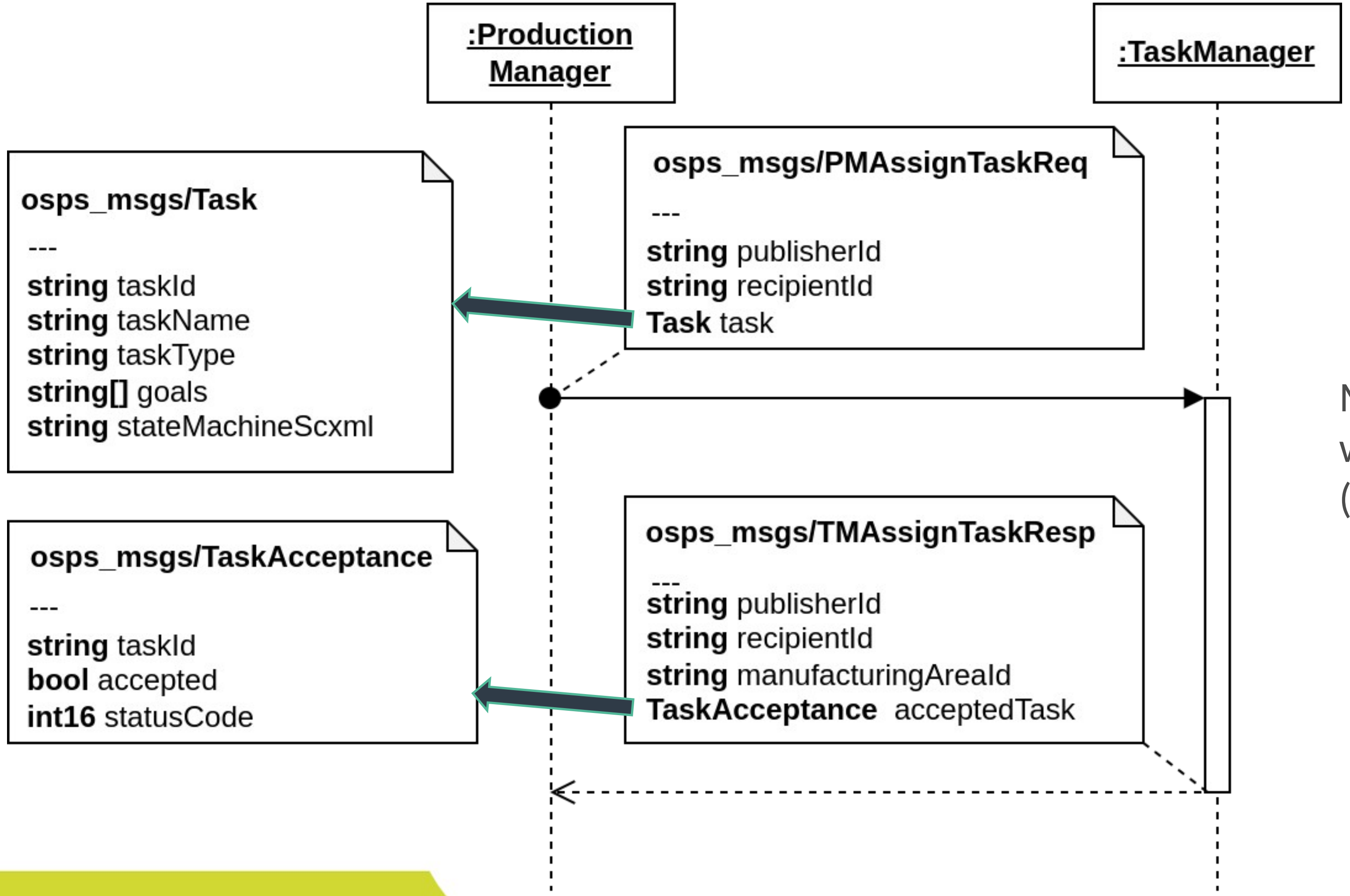
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS Messages – Retrieve robot capabilities



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

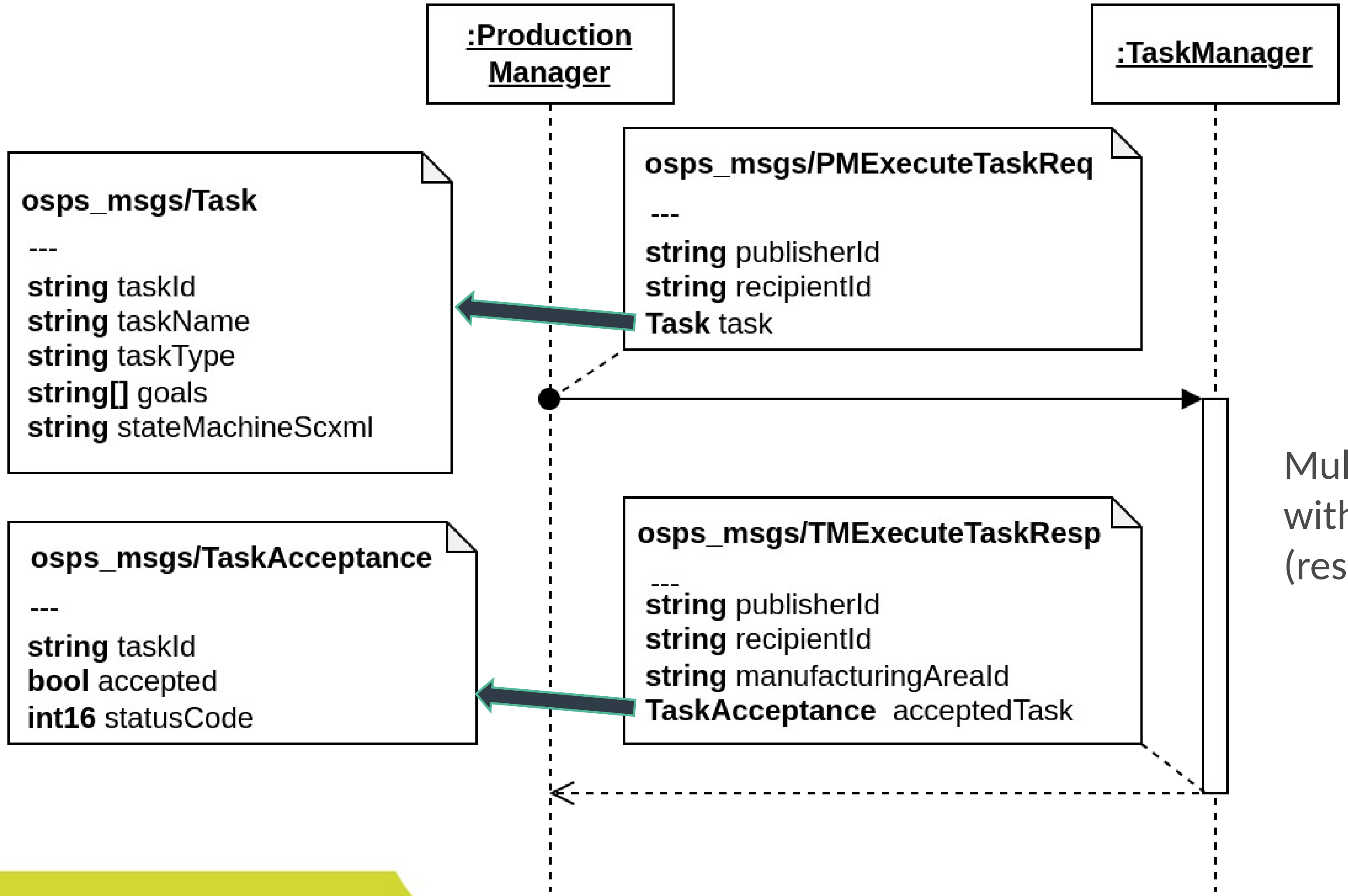
OSPS Messages – Assign a Task for execution



Multiple Tasks may be assign for execution with *osps_msgs/PMAssignTaskListReq* (response is *osps_msgs/TMAssignTaskListResp*)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

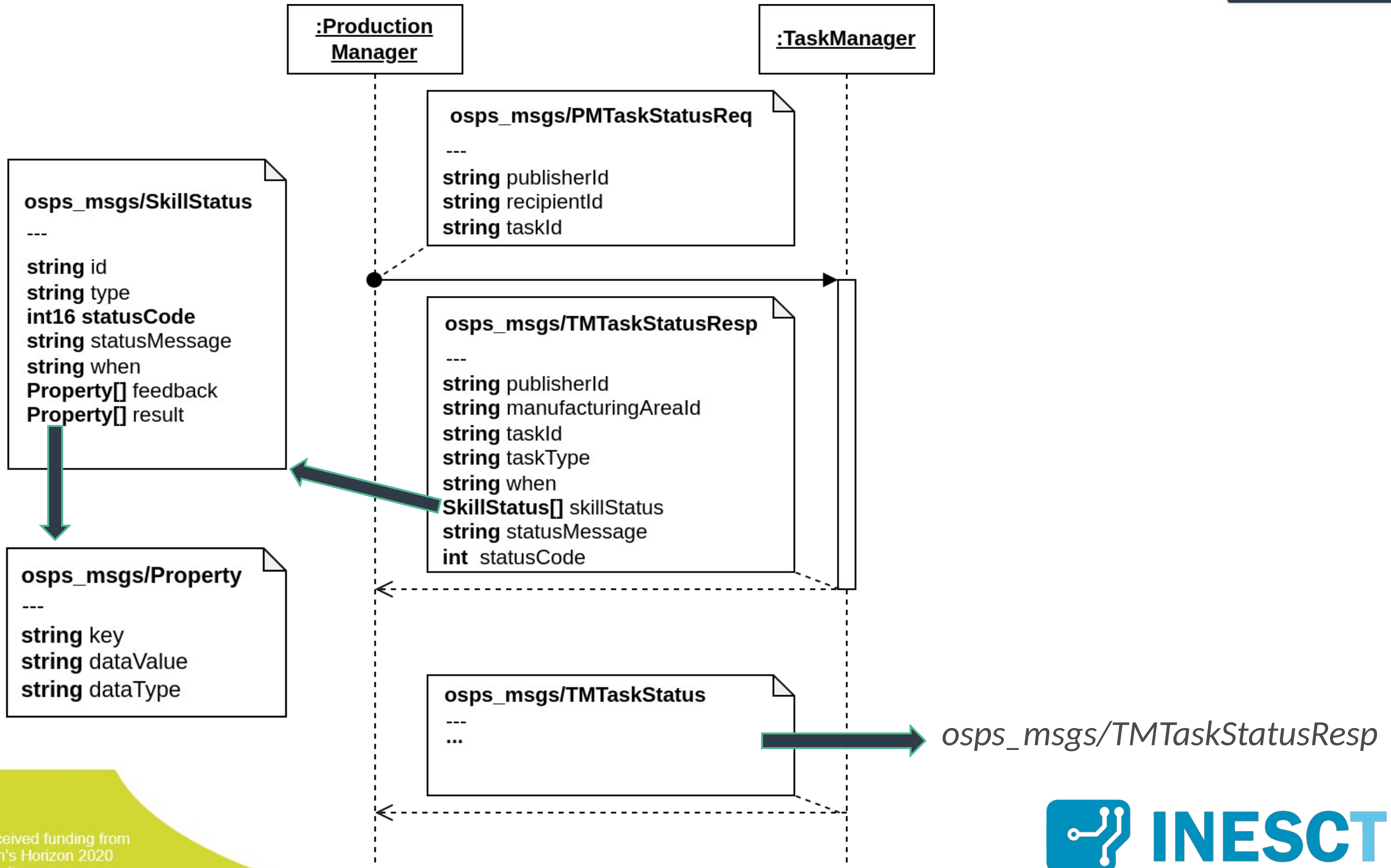
OSPS Messages – Request execution of a Task



Multiple Tasks may be executed (sequentially) with *osps_msgs/PMExecuteTaskListReq* (response is *osps_msgs/TMExecuteTaskListResp*)

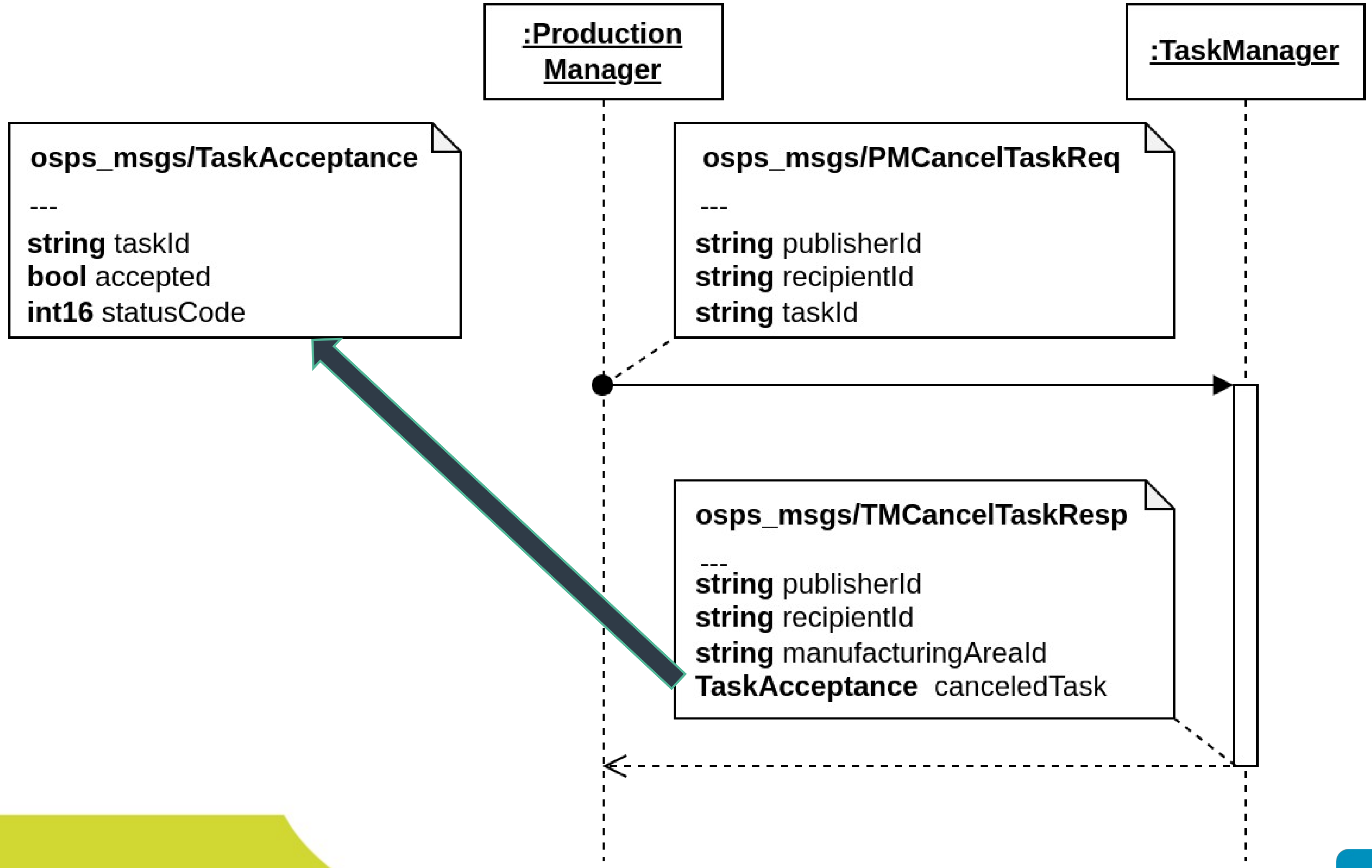
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS Messages –Task execution monitoring



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

OSPS Messages – Cancel Task execution



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Introduction

- Manual generation of skills is labor/time intensive and error-prone;
- Skills share a significant amount of boilerplate code between themselves.

Skill Generator – application that automates the generation of Skill ROS Packages

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Configuration file

The Skill Generator takes as input a YAML configuration file

- **server_language -** support for options ***python*** and *cpp* (client always *Python*)
- **ros_distro -** *melodic*, *noetic* or *foxy* (not yet supported by Production Manager)
- **feedback –** by default it includes the following fields:
  - **percentage** (int32) **-** the percentage of completion
  - **skillStatus** (string) **-** textual information regarding what is being done at a given moment
- **result -** by default it includes the following fields:
  - **percentage** (int32) **-** the final percentage of completion
  - **skillStatus** (string) **-** textual information regarding how the skill finished
  - **outcome** (string) **-** the final skill outcome
- **outcomes -** by default it always includes:
  - **succeeded –** *default* outcome for whenever execution ends with success
  - **aborted –** *default* outcome for whenever an error occurs during execution
  - **preempted –** outcome for whenever the skill execution is cancelled externally

Mandatory Fields


```yaml
skill_name: random_outcome

ros_distro: melodic

server_language: python

goals:
  generic_nr: int32
  generic_str: string
  generic_flt: float
feedback:
  nr_feedback: uint32
  str_feedback: string

result:
  nr_result: uint32
  str_result: string

outcomes:
  - outcome_1
  - outcome_2
```


Optional Fields

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Generating Skill ROS Packages

```
pedro@earth:~/Projects/neoadvance_ws/src/skill_generator(master)$ python3 skill_generator.py generate ../wait_skill.yml
GENERATE MODE:

[WARNING] [YAMLParser] Optional field "feedback" not specified. Default value will be used.
[WARNING] [YAMLParser] Optional field "result" not specified. Default value will be used.
[WARNING] [YAMLParser] Optional field "outcomes" not specified. Default value will be used.
[WARNING] [YAMLParser] Optional field "skill_type" not specified. Default value will be used.
[INFO] [YamlParser] Generating skill folder at "../wait_skill" .
[INFO] [ClientGenerator] Package "wait_skill_client" generated with success.
[INFO] [MsgsGenerator] Package "wait_skill_msg" generated with success.
[INFO] [ServerGenerator] Package "wait_skill_server" generated with success.
[DEBUG] [SCXMLGenerator] Obtaining ROS1 messages.
[INFO] [SCXMLGenerator] Package "wait_skill_scxml" generated with success.

Generation completed in 0.7029163837432861 seconds.
```

The following folders are created:

- X_skill_client: contains code for the ROS Action client
- X_skill_msg: contains the required ROS .action files
- X_skill_scxml: contains a sample SCXML file

- **X_skill_server: ROS package - contains code for the ROS Action server**

```
├─ wait_skill
   ├─ wait_skill.yaml                          Input yaml file
   ├─ wait_skill_client                        Skill Client
   │   ├─ CMakeLists.txt
   │   ├─ package.xml
   │   ├─ setup.py
   │   └─ src
   │       └─ wait_skill_client
   │           ├─ __init__.py
   │           └─ wait_skill_client.py
   ├─ wait_skill_msgs                          Skill Msgs
   │   ├─ action
   │   │   └─ WaitSkill.action
   │   ├─ CMakeLists.txt
   │   └─ package.xml
   ├─ wait_skill_scxml                         Skill scxml file
   │   └─ wait_skill.scxml
   └─ wait_skill_server                        Skill Server Implementation of the skill
       ├─ CMakeLists.txt
       ├─ launch
       │   └─ run.launch
       ├─ package.xml
       ├─ scripts
       │   └─ wait_skill.py
       ├─ setup.py
       ├─ src
       │   └─ wait_skill_server
       │       ├─ __init__.py
       │       └─ wait_skill_class.py
       └─ test
           ├─ integration_test.test
           ├─ test_wait_skill_class.py
           ├─ unit_test.test
           └─ wait_skill_server_integration_tests.py
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Skill Server API* (Python)

* Function arguments may change depending on the Skill configuration. Only default arguments are shown.

__init__(action_name: str = 'XSkill') -> None	**Constructor - Initialize global state**
execute_skill(goal: WaitSkillAction) -> None	**Start executing skill logic**
aborted(status: str = None, outcome='aborted') -> None **success**(status: str = None, outcome='suceeded') -> None	**Stop executing skill logic and provide final result with outcome**
check_preemption() -> bool	**Verify if the skill execution was preempted by the Task Manager**
feedback(status: str = None) -> None	**Verify if the skill execution was preempted by the Task Manager**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Skill Server API* (C++)

* Function arguments may change depending on the Skill configuration. Only default arguments are shown.

ExampleSkill(std::string name)	**Constructor - Initialize global state**
void **executeCB**(const *example_skill_msgs*::ExampleSkillGoalConstPtr &*goal*)	**Start executing skill logic**
void **set_aborted**(std::string outcome="aborted") void **set_succeeded**(std::string outcome="succeeded")	**Stop executing skill logic and provide final result with outcome**
void **check_preemption**()	**Verify if the skill execution was preempted by the Task Manager**
void **feedback**(int percentage)	**Verify if the skill execution was preempted by the Task Manager**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Implementing Functionality (Example)

```python
import rospy
import actionlib
from wait_skill_msgs.msg import WaitSkillAction, WaitSkillResult, WaitSkillFeedback

class WaitSkill(object):

    def __init__(self, action_name='WaitSkill'):▬

    def execute_skill(self, goal):
        '''
        The execution of the skill should be coded here.In order to save you time,
        the methods check_preemption(), feedback(), success() and aborted() should be used.
        The check_preemption() method should be called periodically.
        The variable self.percentage should be updated when there is an evolution
        in the execution of the skill.
        The feedback() method should be called when there is an evolution in the
        execution of the skill.
        '''

    def feedback(self, status=None):▬

    def success(self, status=None, outcome='succeeded'):▬

    def aborted(self, status=None, outcome='aborted'):▬

    def check_preemption(self):▬

    def result_constructor(self, status, percentage=None, outcome=None):▬

    @staticmethod
    def log_info(status):▬
```

Default *wait_skill* server generated by the Skill Generator

```python
import rospy
import actionlib
import time
from wait_skill_msgs.msg import WaitSkillAction, WaitSkillResult, WaitSkillFeedback

class WaitSkill(object):

    def __init__(self, action_name='WaitSkill'):▬

    def elapsed_time(self):▬

    def execute_skill(self, goal):

        self.start_time = time.time() # Sets starting time as current time

        while not self.check_preemption() : # While not preempted
            if self.elapsed_time() < goal.waitTime : # Waits until the time in goal passes

                skillStatus = 'Elapsed Time: ' + str(
                round(self.elapsed_time())) + 's. Remaining Time: ' + str(
                round(goal.waitTime - self.elapsed_time())) + 's' # Skill Status

                self.feedback(skillStatus) # Skill feedback

                # Defining percentage of the skill done
                self.percentage = int(round(self.elapsed_time() / goal.waitTime * 100))
                time.sleep(1.0)

            else : # If skill terminates normally sets success and breaks loop.
                self.success('Waited successfully ' + str(goal.waitTime) + 's')
                break

        # success(), aborted(), check_preemption(), result_constructor() and log_info()
```

Implementation of the *wait_skill* server

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Skill Client API

Seldomly used directly by developers!

```
import sys
import rospy

from task_manager_server.skill_class import SkillSetup, SkillExecution, SkillAnalysis
```

```
class WaitSkillSetup(SkillSetup):
    pass
```

Allows overloading methods from the default client
action name, action goal, action type and result constructors

```
class WaitSkillExecution(SkillExecution):
    pass
```

Allows overloading methods from the default client
action feedback, action done, action active callbacks

```
class WaitSkillAnalysis(SkillAnalysis):
    pass
```

Discontinued functionality

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Update Skill ROS Package

Changes to existing Skill ROS packages must be done manually!

- Always keep your original configuration YAML file updated for future reference and documentation

Outcomes:

- Update list of outcomes in the Skill client XSkillAnalysis (x_skill_clie*nt.py*)
- **[Python]** Alter the Skill server so that XSkill class variable *self.outcomes* includes the custom outcomes (x_skill_server.py)

```
class ExampleSkillAnalysis(SkillAnalysis):
    def set_outcomes(self, outcomes=None):
        return ['preempted', 'aborted', 'succeeded', "outcome_A", "outcome_B", "outcome_C"]
```

Result & Feedback:

- Update ROS .action file
- **[C++]** Alter server functions *set_succeeded*, *set_aborted*, and *feedback* as to accept as arguments all required fields
- **[Python]** Alter server functions *success*, *aborted*, and *feedback* as to accept as arguments all required fields

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Skill Generator – Launch Skill Server

```
pedro@earth:~$ roslaunch wait_skill_server run.launch
... logging to /home/pedro/.ros/log/f1c67986-3041-11ed-a080-33a4031e63fb/roslaunch-earth-34745.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://earth:44485/

SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.14
 * /wait_skill/action_name: WaitSkill

NODES
  /
    wait_skill (wait_skill_server/wait_skill.py)

auto-starting new master
process[master]: started with pid [34754]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to f1c67986-3041-11ed-a080-33a4031e63fb
process[rosout-1]: started with pid [34773]
started core service [/rosout]
process[wait_skill-2]: started with pid [34776]
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Manager

Configuration and Launching

This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Task Manager - Configuration

Task Manager obtains basic robot information about a static YAML file
(*task_manager_scxml_stack/task_manager/config/robot_configuration.yaml*)

```yaml
robot_configuration:
  robot_id: 'friday'

  robot_type: 'Robot.MobileManipulator'

  robot_volume: {height: 1,
                 polyLine: [{x: 0, y: 0, z: 0},
                            {x: 0, y: 1, z: 0},
                            {x: 1, y: 1, z: 0},
                            {x: 1, y: 0, z: 0}]}

  # manufacturing_area_id: 'PhysicalArea'
  manufacturing_area_id: 'CRIIS_ASSEMBLY_LINE'

  properties:
    - {key: 'default_property1', dataValue: 'value1', dataType: 'type1'}

  bag: False # set to True to automatically record a .bag file whenever a task is executed
  docker: False # set to True if this robot is running inside a docker container
```

Unique robot identifier (robot_id)

Robot area identifier (manufacturing_area_id)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Manager - Launch

```
pedro@earth:~$ roslaunch task_manager run.launch
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://earth:42709/

NODES
  /
    task_manager (task_manager_server/task_manager_server_node.py)
    task_manager_heart_beep (task_manager_heart_beep/heart_beep_node.py)
    task_manager_robot_map (task_manager_robot_map/robot_map_node.py)
    task_manager_robot_profile (task_manager_robot_profile/robot_profile_node.py)

auto-starting new master
process[master]: started with pid [11059]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 6dab363a-3804-11ed-a520-b76e29696638
process[rosout-1]: started with pid [11069]
started core service [/rosout]
process[task_manager-2]: started with pid [11073]
process[task_manager_heart_beep-3]: started with pid [11077]
process[task_manager_robot_profile-4]: started with pid [11078]
process[task_manager_robot_map-5]: started with pid [11079]
[INFO] [1663582720.226690]: [friday] [TaskManager] [RobotMap] - Ready to receive Robot Map requests
[INFO] [1663582720.248892]: [friday] [TaskManager] [HeartBeep] - Ready to publish heart beep
WARNING: topic [/IOT/WifiInfo] does not appear to be published yet
[WARN] [1663582720.290795]: [friday] [TaskManager] - Found Robot Skills: ['wait_skill']
[INFO] [1663582726.256748]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582728.265302]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582730.270166]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582732.272900]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582734.278357]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582736.283768]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
[INFO] [1663582738.292668]: [friday] [TaskManager] [HeartBeep] - Using default/empty transform in HeartBeep
```

Skills detected

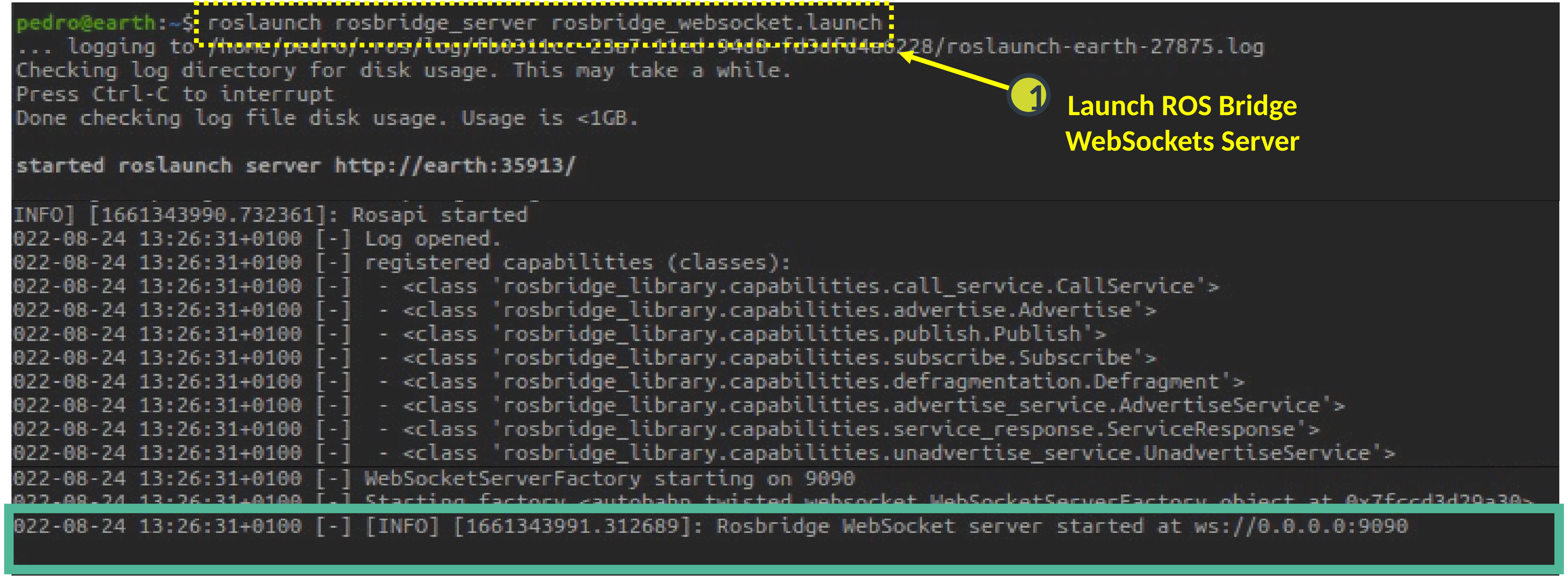
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager

Connection to robots and monitoring

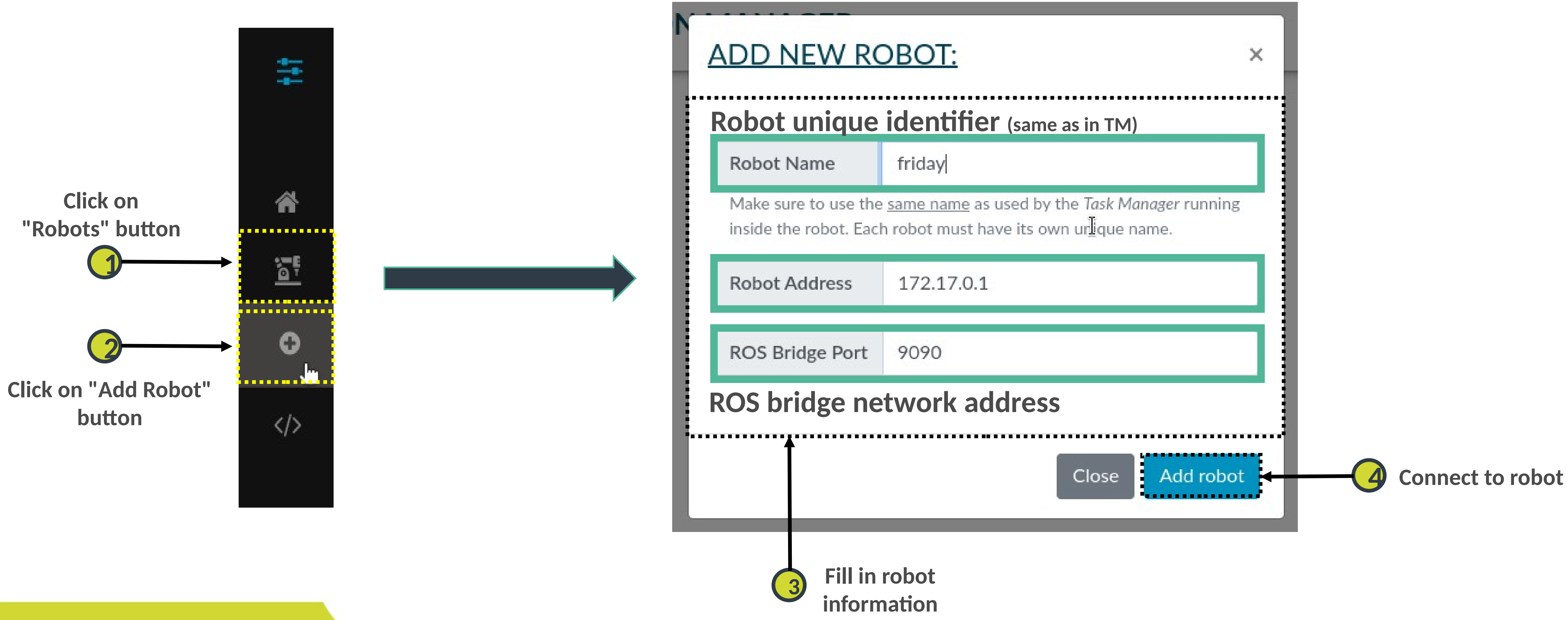
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Launch ROS Bridge WebSockets server



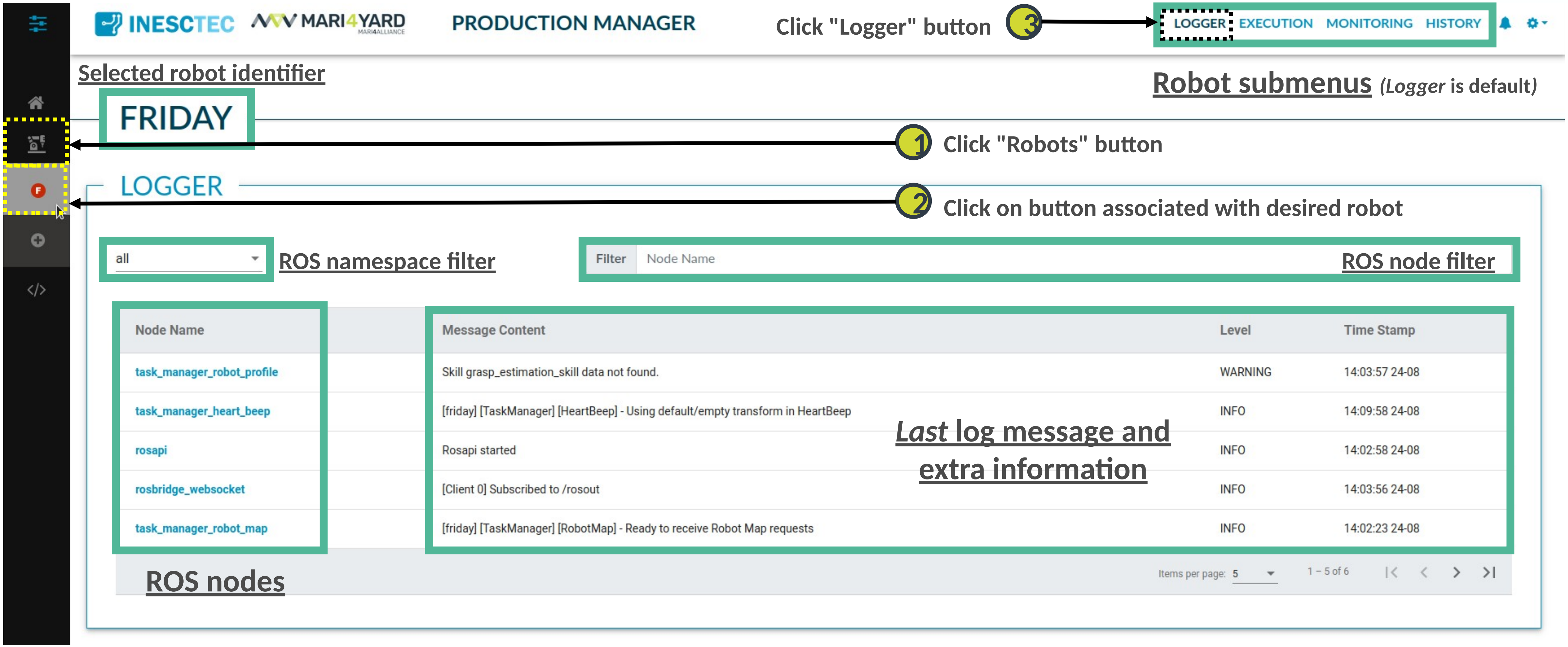
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Connect to a new robot



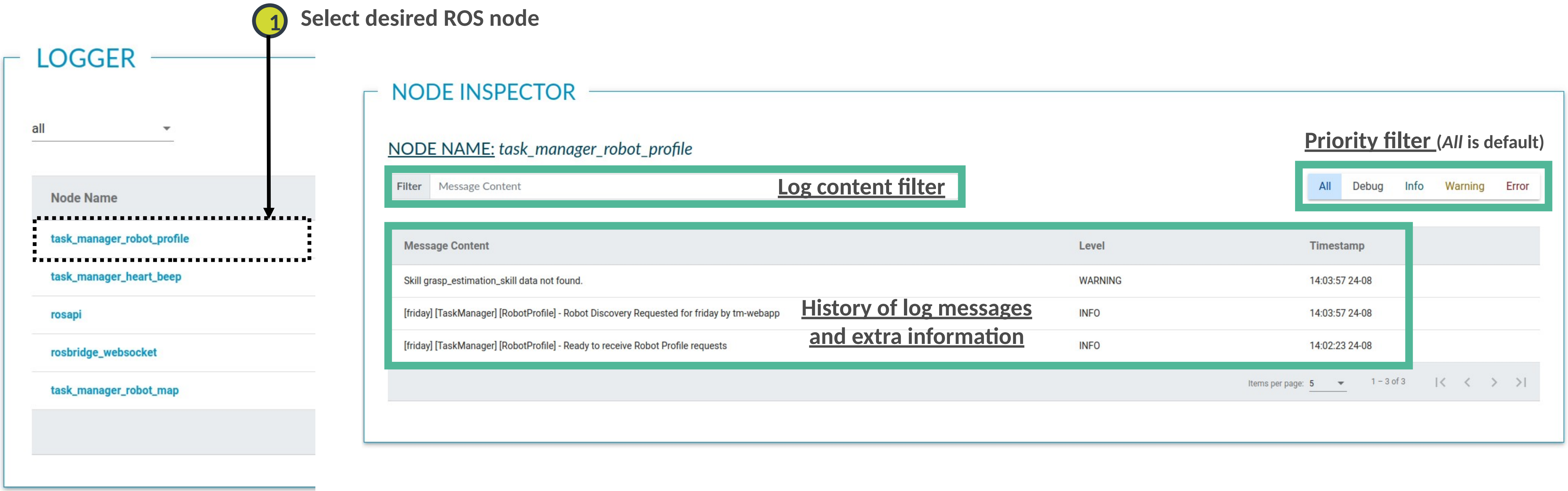
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Robot Homepage / Logger Submenu



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Inspect Robot Logs



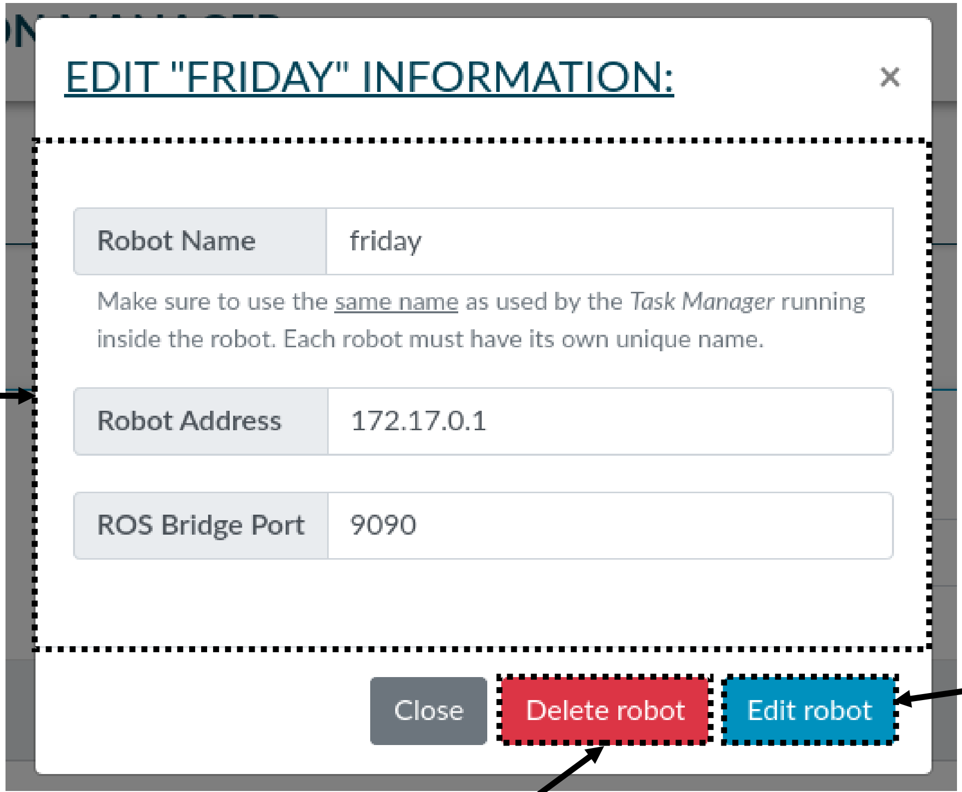
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Edit Robot Information / Delete Robot

1 Click "Robot Settings" button

2 Click "Edit Robot" button

3 Update robot information



4 Click "Edit Robot" to update information

4 Update robot and all related information

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Robot Homepage / Logger Submenu

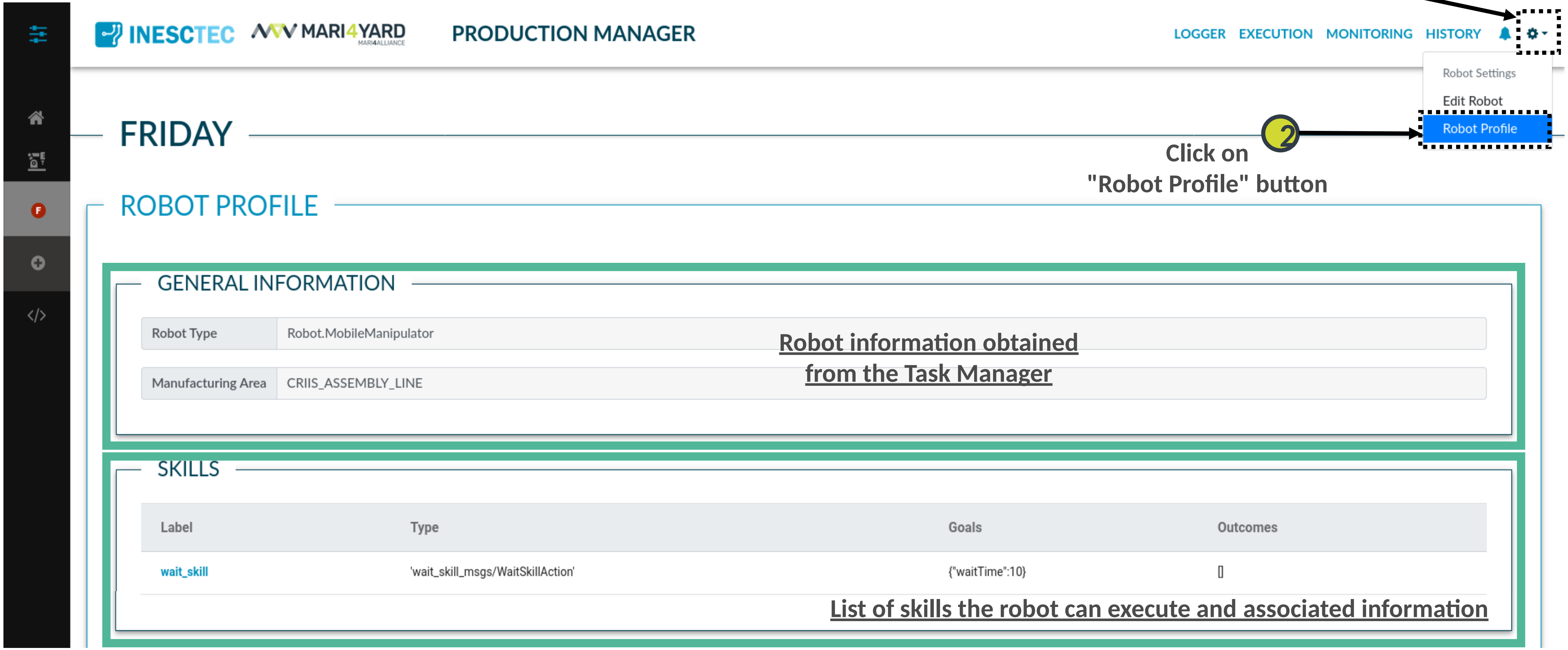
ROS Bridge automatically detects connection from Production Manager

```
2022-08-24 14:02:58+0100 [-] WebSocketServerFactory starting on 9090
2022-08-24 14:02:58+0100 [-] Starting factory <autobahn.twisted.websocket.WebSocketServerFactory object at 0x7f1f2ec04880>
2022-08-24 14:02:58+0100 [-] [INFO] [1661346178.697776]: Rosbridge WebSocket server started at ws://0.0.0.0:9090
2022-08-24 14:03:56+0100 [-] <autobahn.websocket.protocol.WebSocketServerProtocol.onConnect>: request={'extensions': [('permessage
-deflate', {'client_max_window_bits': [True]})],
         'headers': {'connection': 'Upgrade',
                     'host': '172.17.0.1:9090',
                     'sec-websocket-extensions': 'permessage-deflate; '
                                                 'client_max_window_bits',
                     'sec-websocket-key': 'y3QTRpMuNVLUjCyX0fcn2A==',
                     'sec-websocket-version': '13',
                     'upgrade': 'websocket'},
         'host': '172.17.0.1',
         'origin': '',
         'params': {},
         'path': '/',
         'peer': 'tcp4:172.26.0.8:43444',
         'protocols': [],
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.264113]: Client connected.  1 clients total.
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.509097]: [Client 0] Subscribed to /rosout
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.546969]: [Client 0] Subscribed to /OSPS/TM/TaskStatus
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.552297]: [Client 0] Subscribed to /OSPS/TM/RobotProfile
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.557616]: [Client 0] Subscribed to /OSPS/TM/HeartBeep
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.562348]: [Client 0] Subscribed to /OSPS/PM/ExecuteTaskReq
2022-08-24 14:03:56+0100 [-] [INFO] [1661346236.566548]: [Client 0] Subscribed to /OSPS/TM/TaskContextModelReq
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager - Robot Profile

Click on "Settings" button (1)



Click on "Robot Profile" button (2)

Robot information obtained from the Task Manager

List of skills the robot can execute and associated information

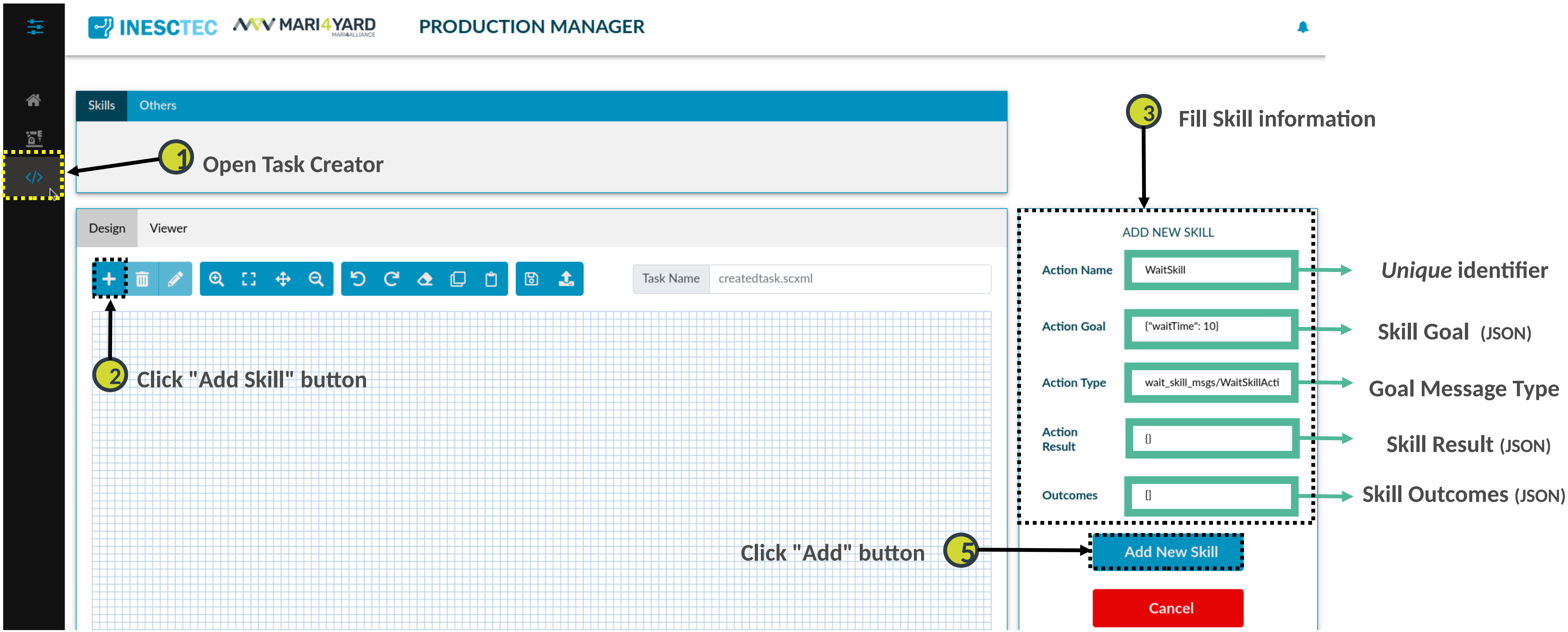
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager

Task Creator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

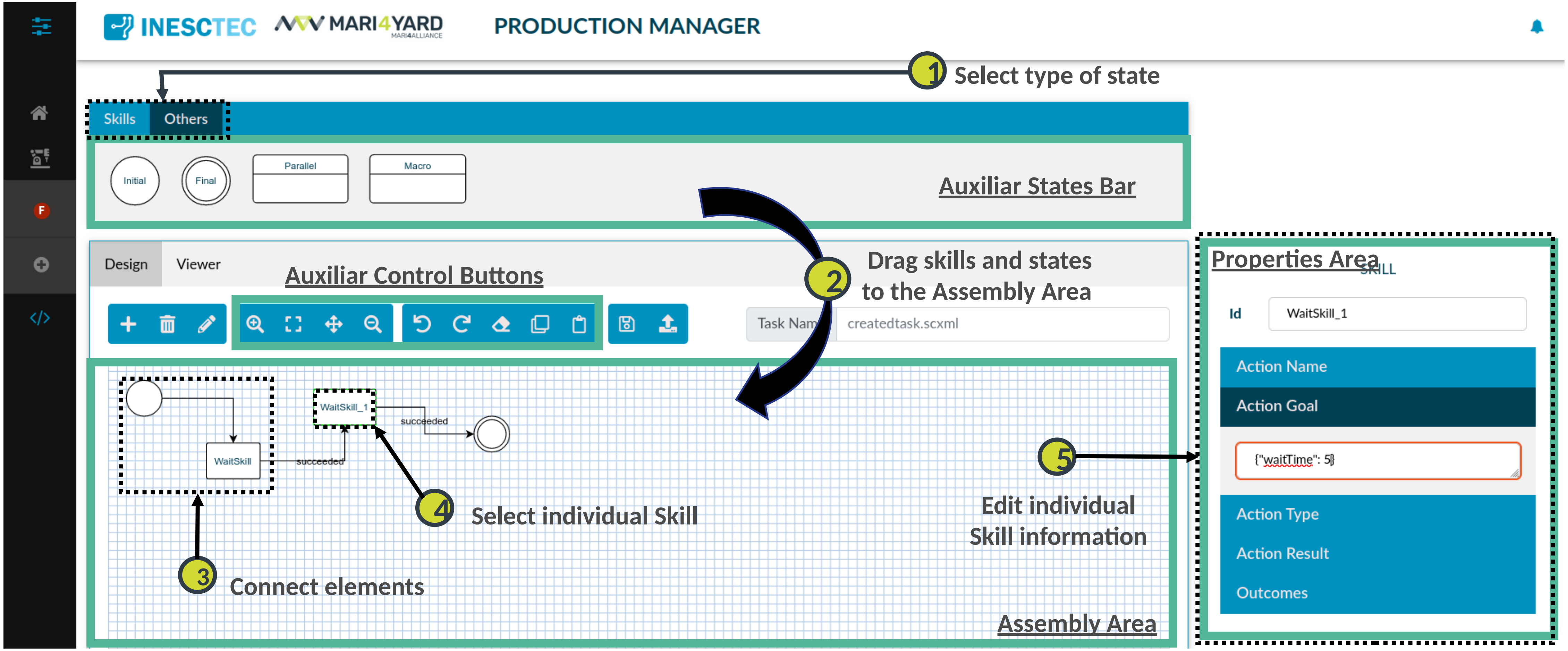
Task Creator – Add New Skill



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

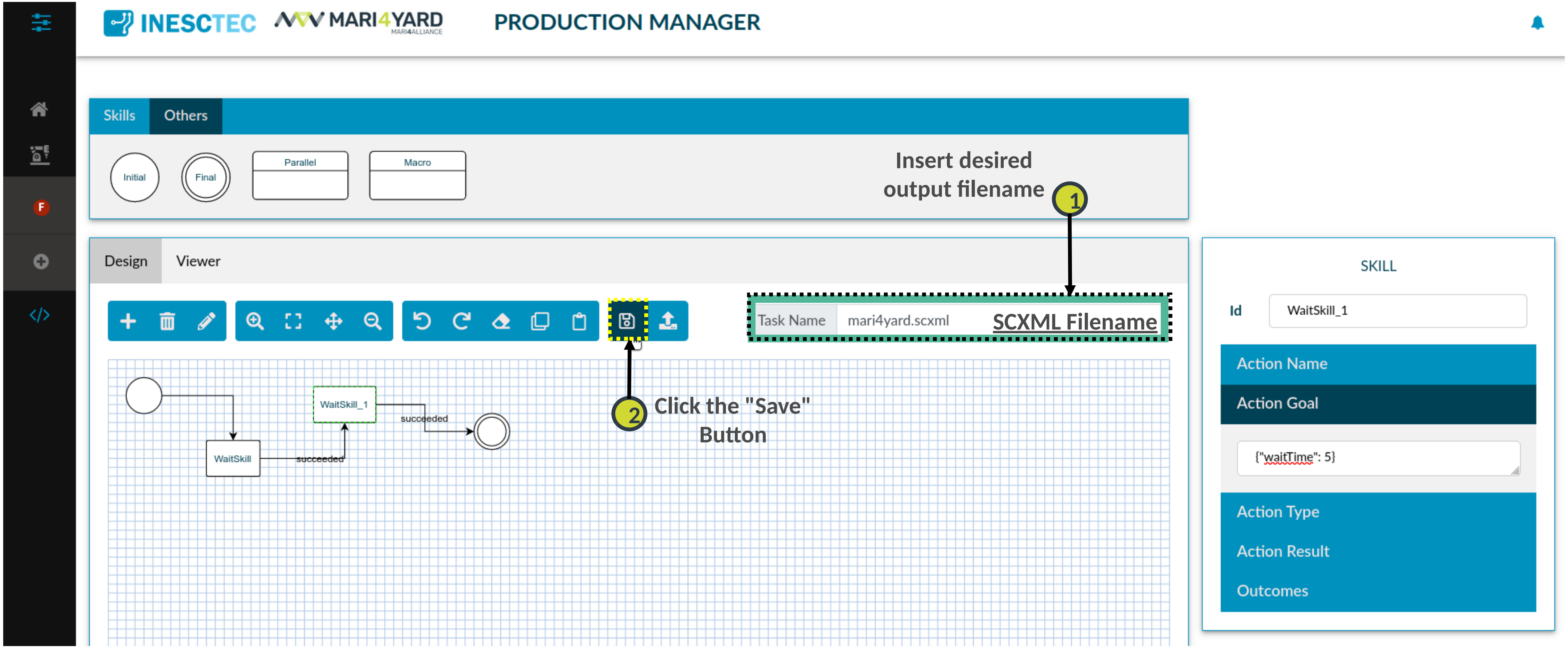
INESCTEC

Task Creator – Assemble Task



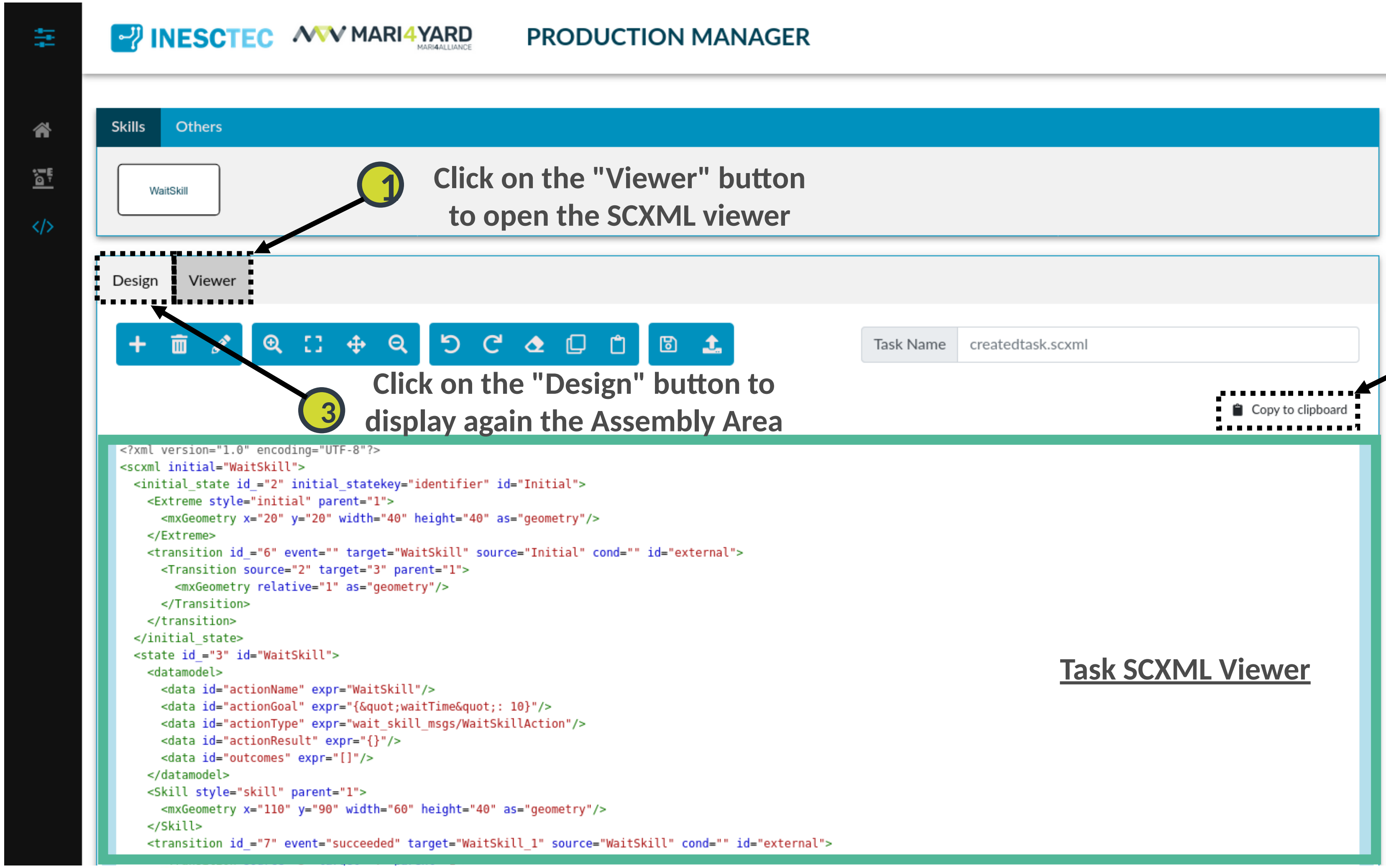
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Creator – Save Task SCXML



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Creator – View/Copy Task SCXML

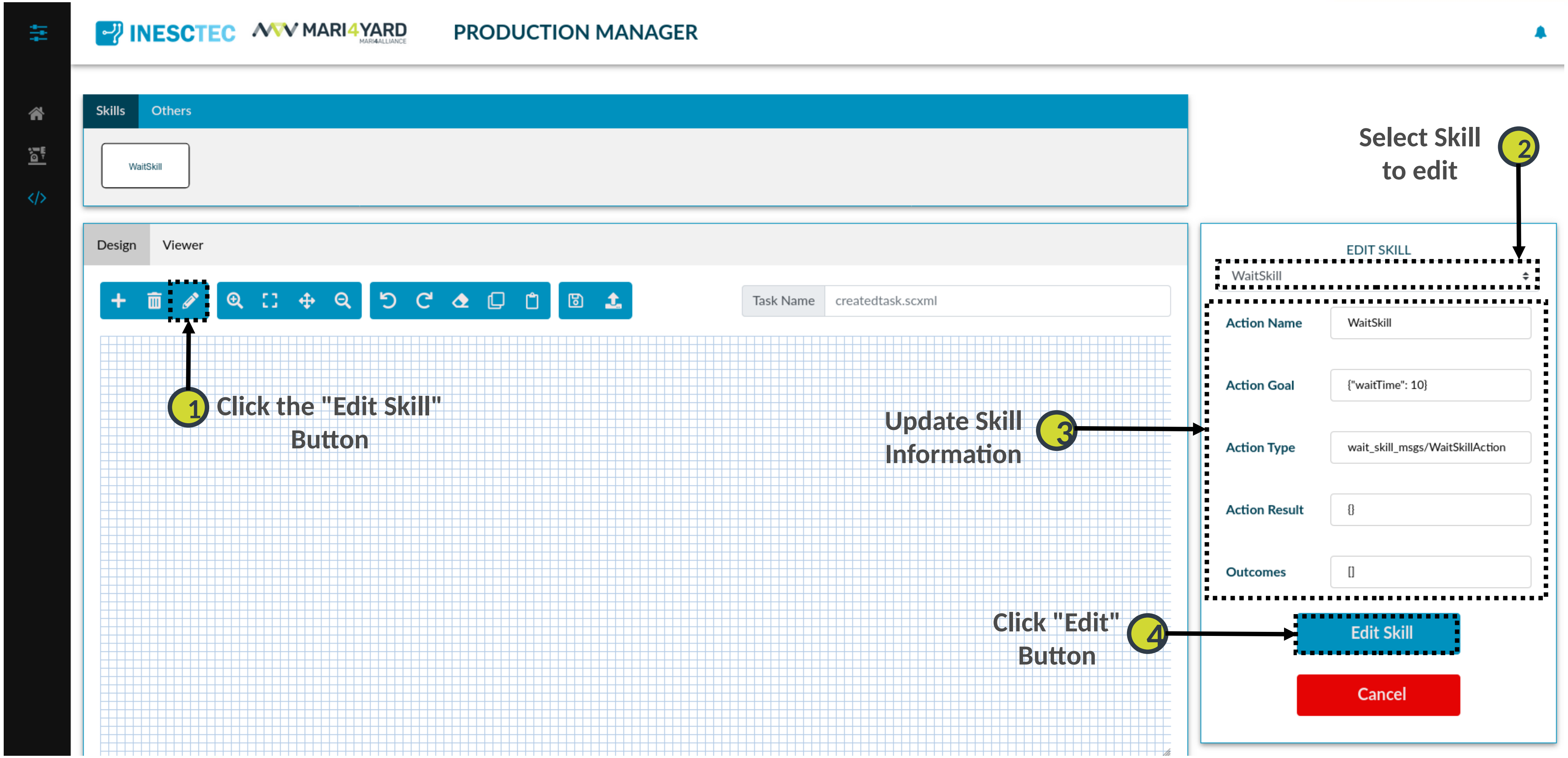


2

Copy the
entire Task's SCXML for
later use

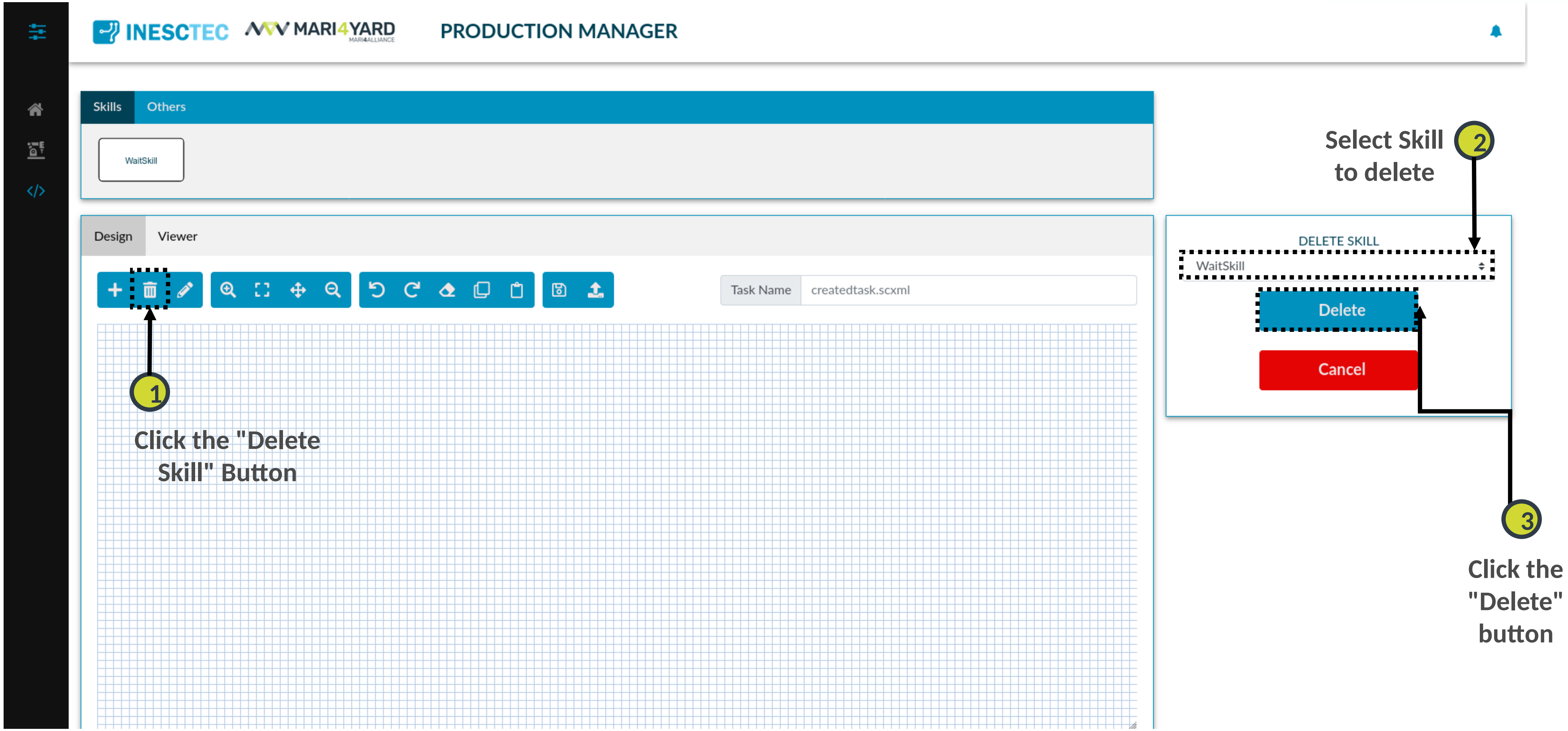
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Creator – Edit Skill



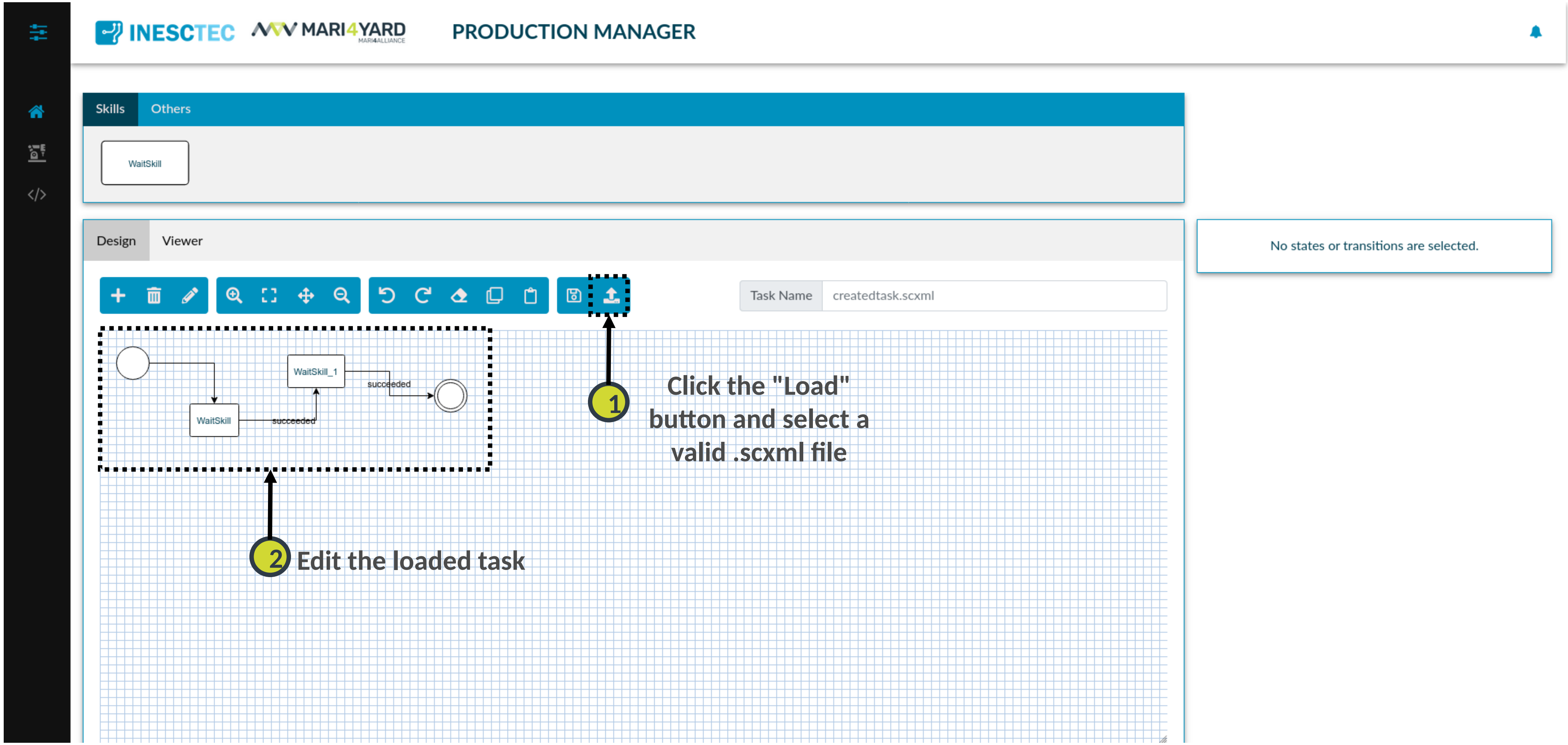
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Creator – Delete Skill



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Task Creator – Edit Task



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

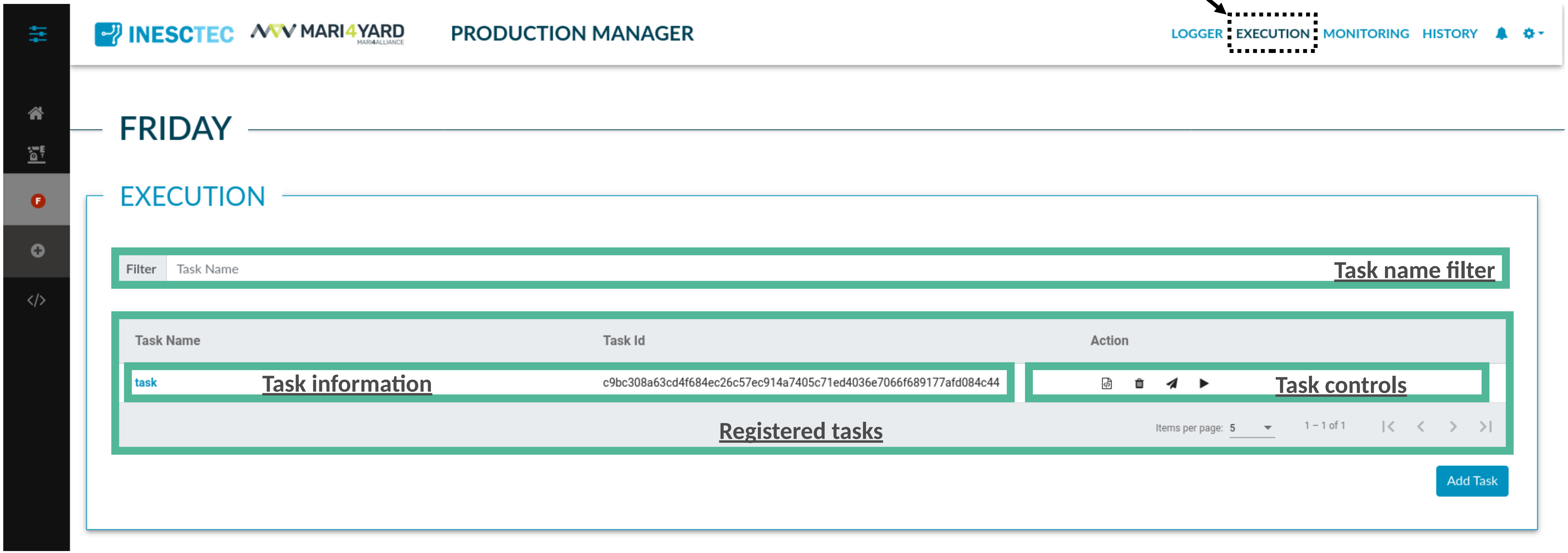
Production Manager

Task Execution

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

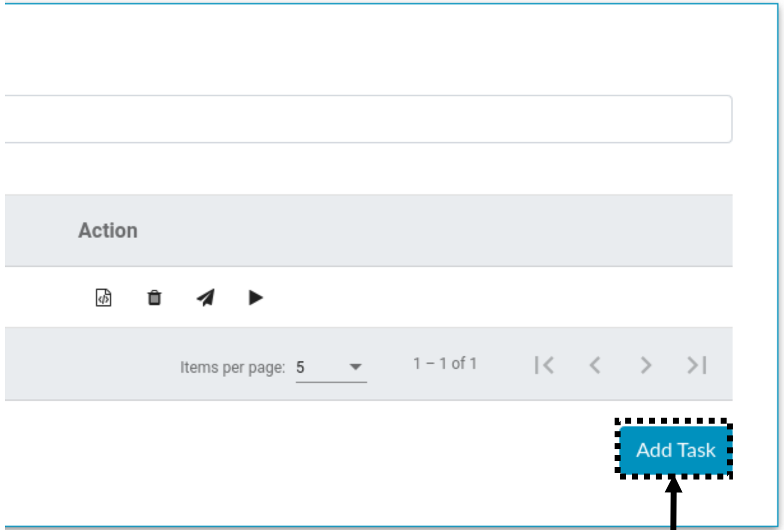
Production Manager – Tasks Menu

Click the "Execution" button
to open the Tasks Menu 1



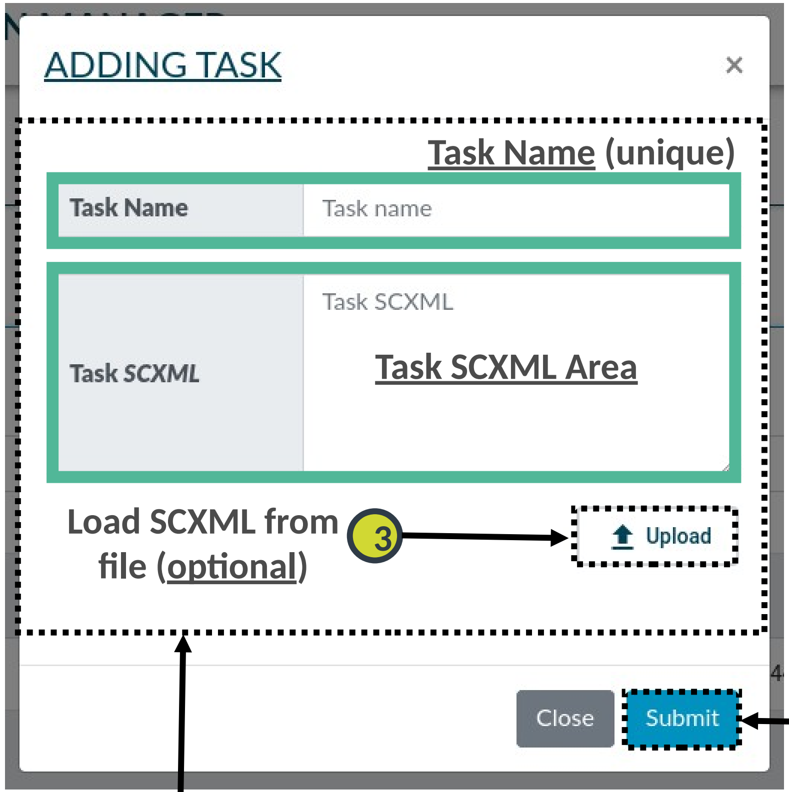
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager – Add Task



1

Click the "Add Task" button



2

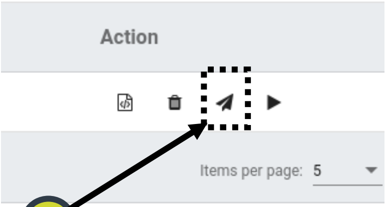
Fill Task information

4 Add Task

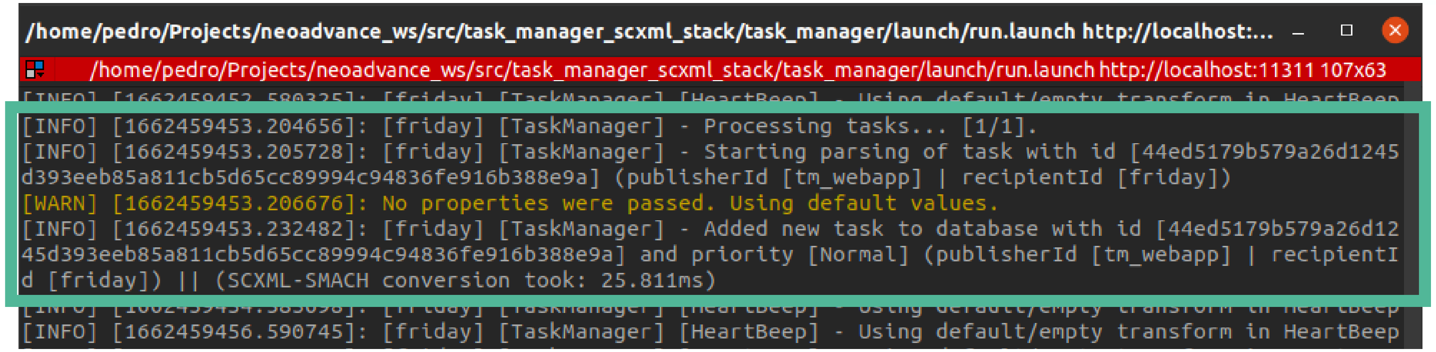
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

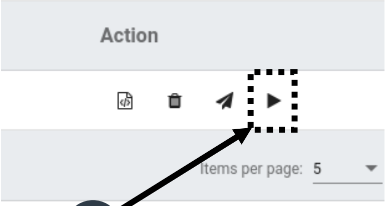
INESCTEC

Production Manager – Execute Task

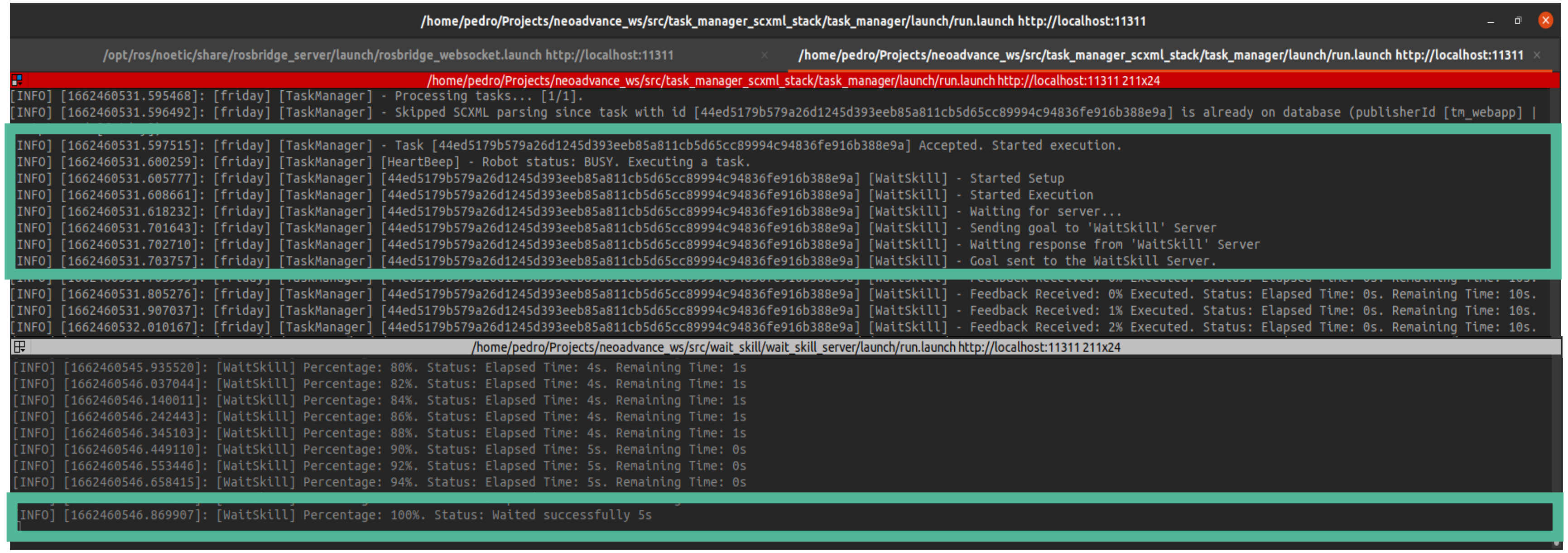


Send *Assign Task* request



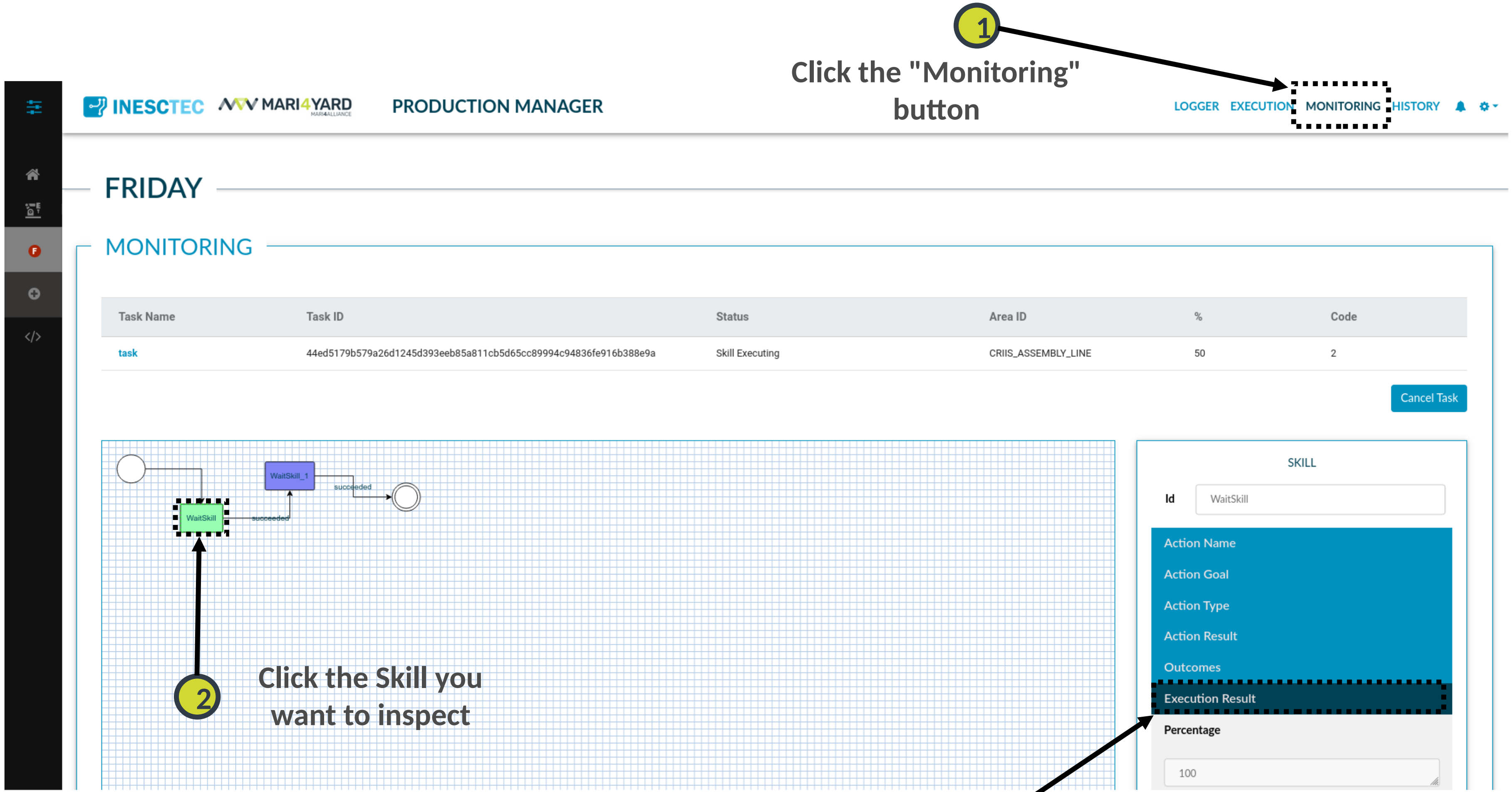


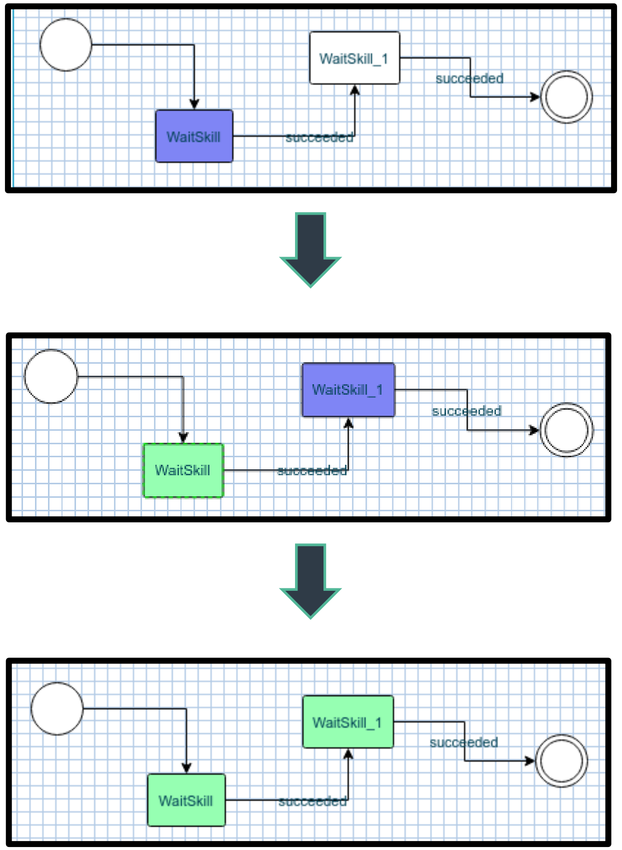
Send *Execute Task* request



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager – Task Execution Monitoring





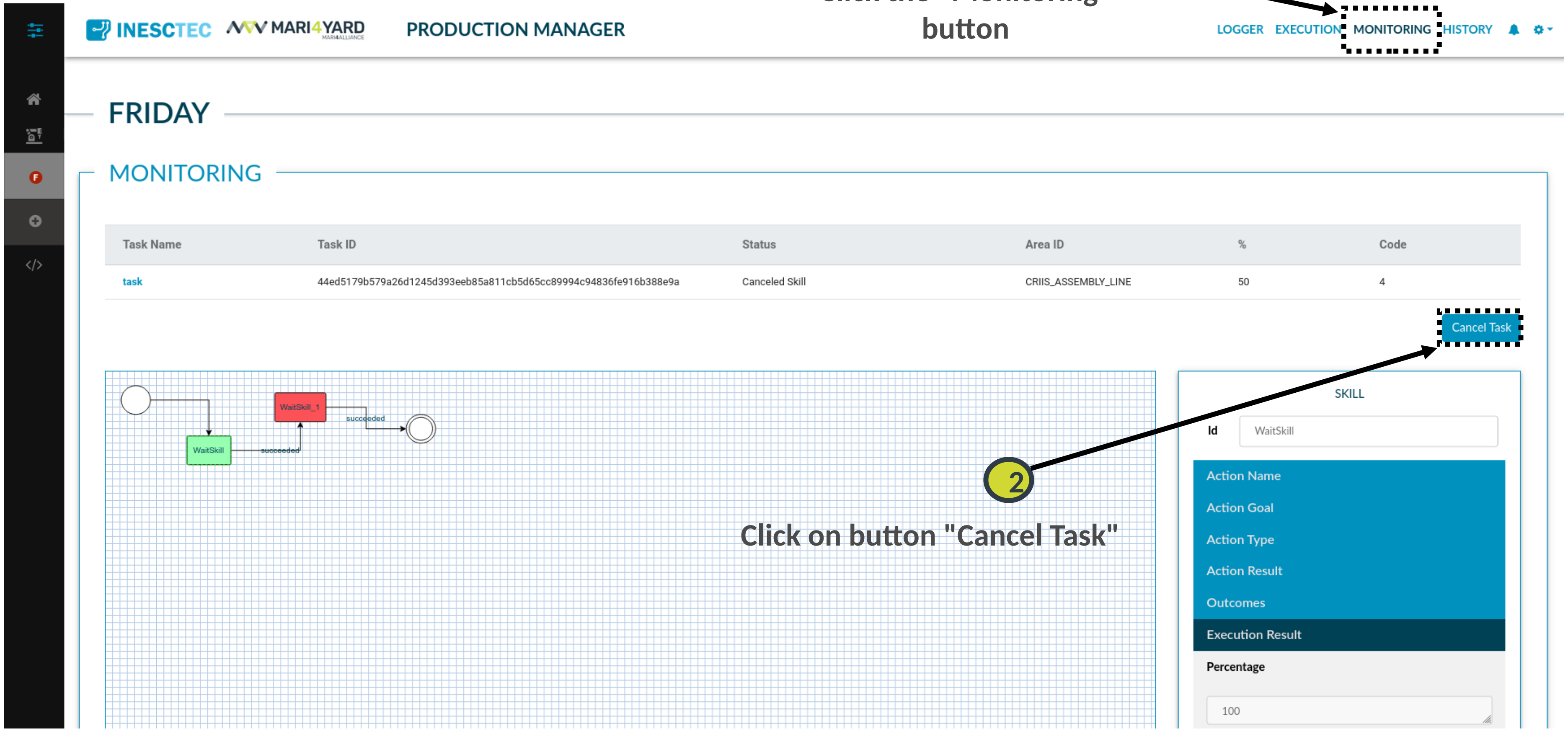
3
Select desired Skill information (only most recent information is kept)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

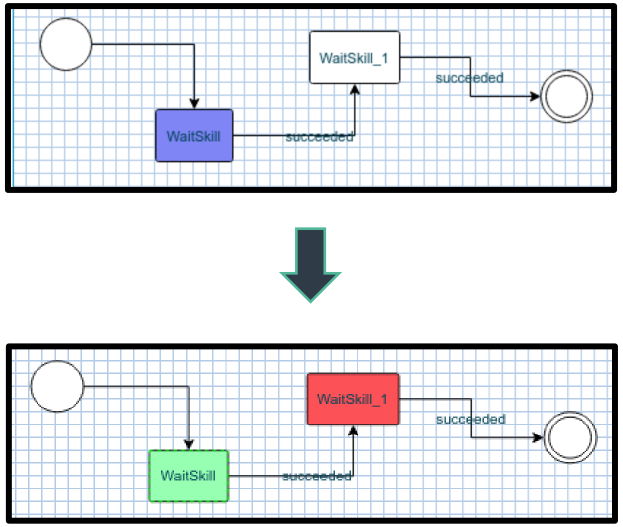
INESCTEC

Production Manager – Cancel Task Execution

1 Click the "Monitoring" button

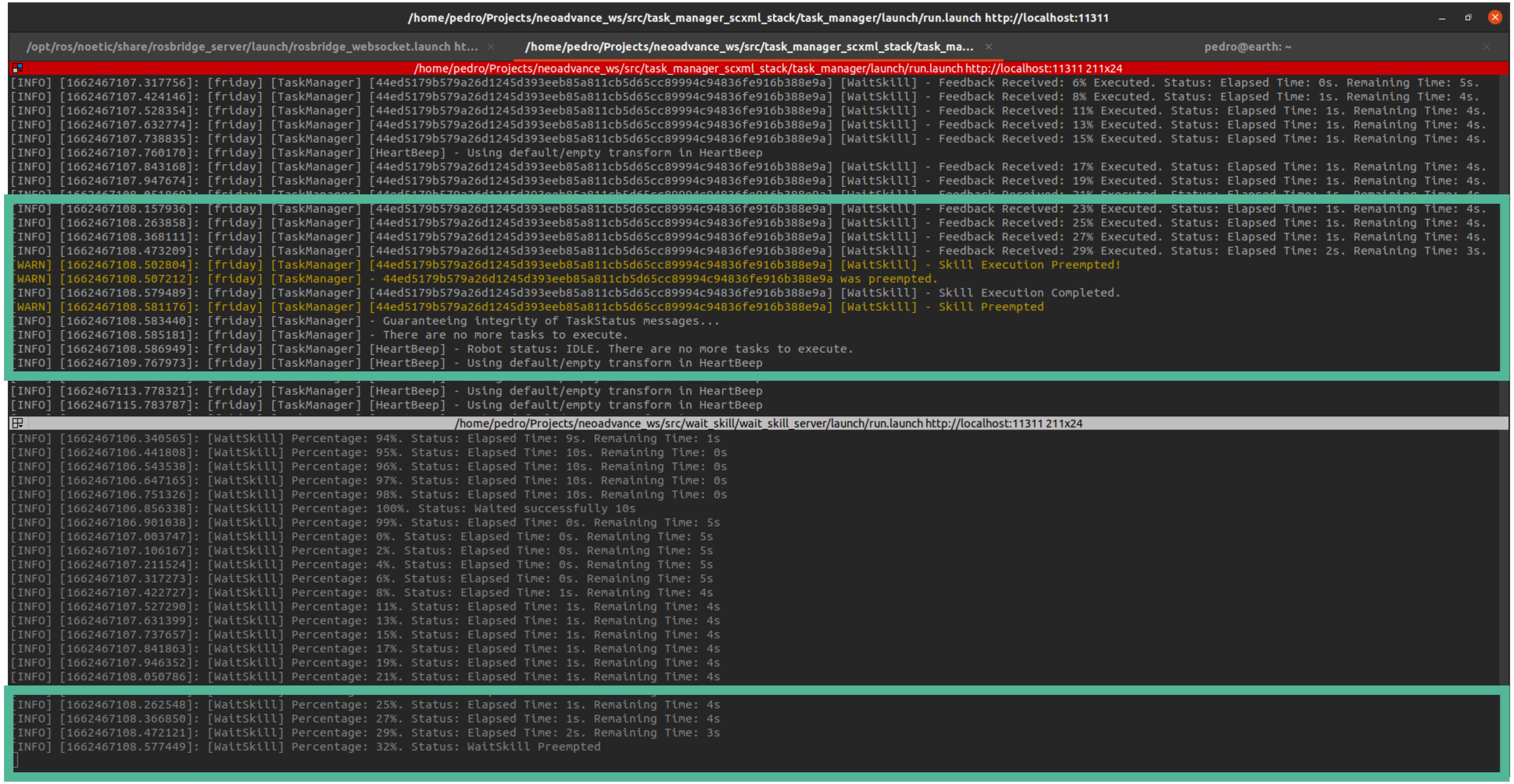


2 Click on button "Cancel Task"



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

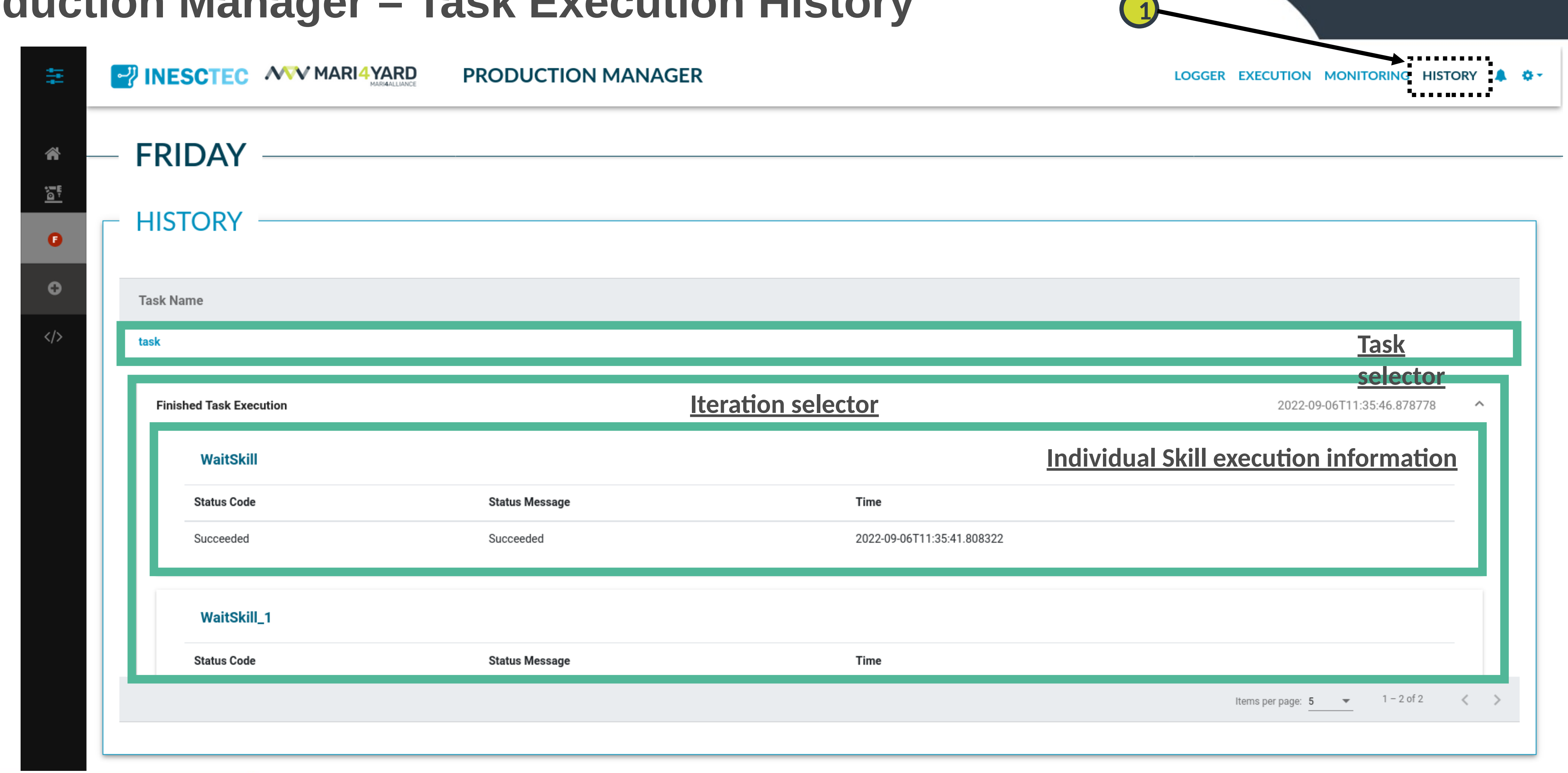
Production Manager – Cancel Task Execution



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

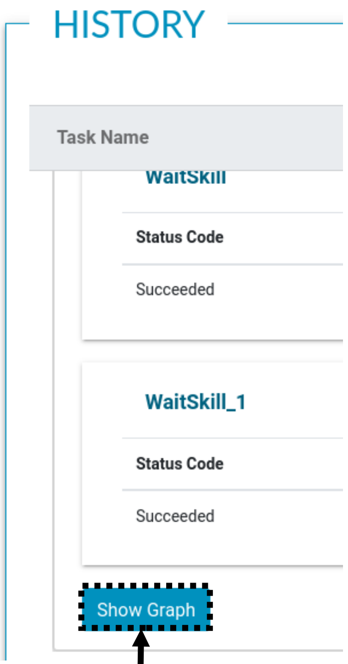
Production Manager – Task Execution History

Click the "History" button



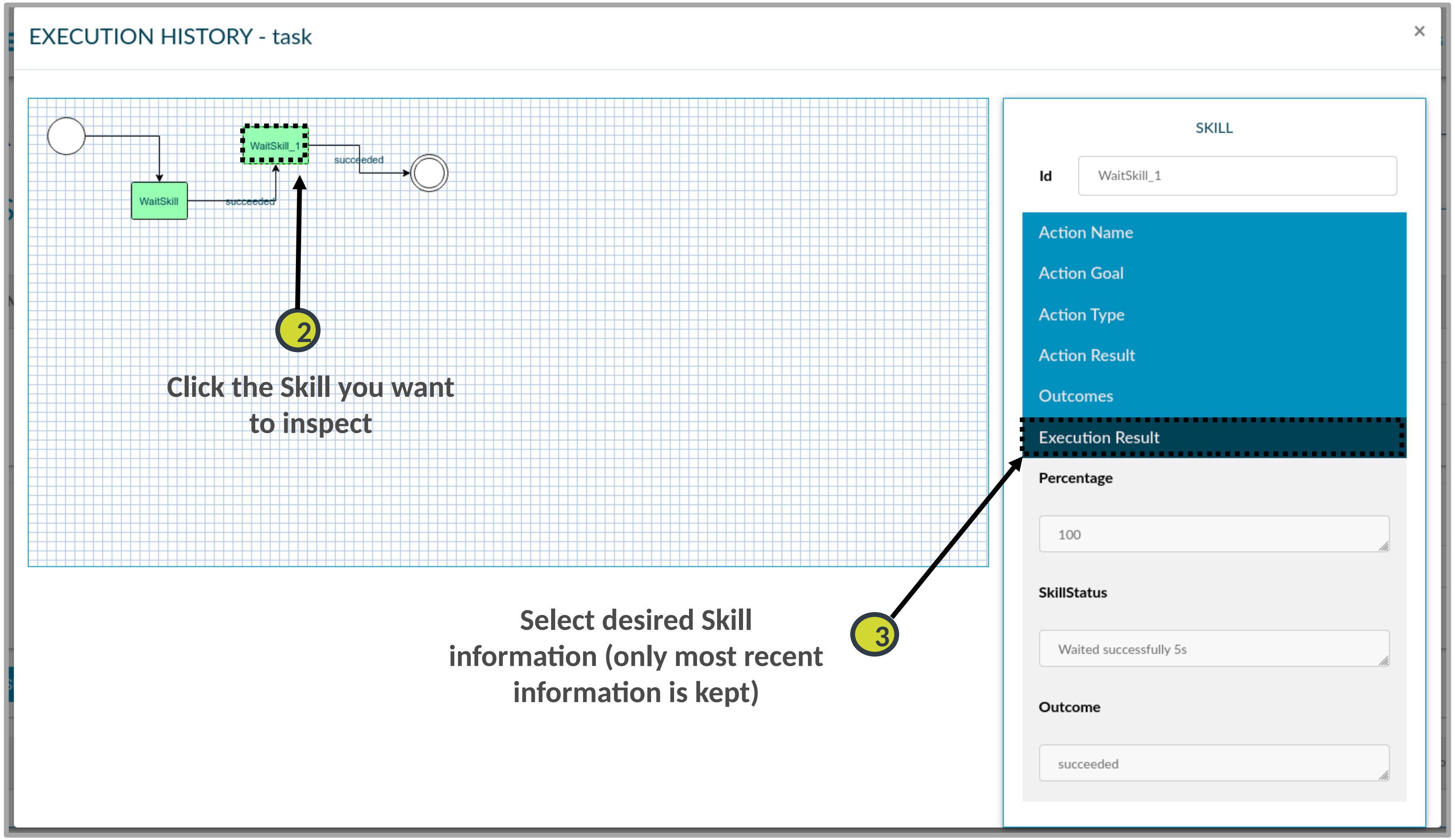
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager – Task Execution History



1

Scroll to the bottom and click on the "Show Graph" button



2

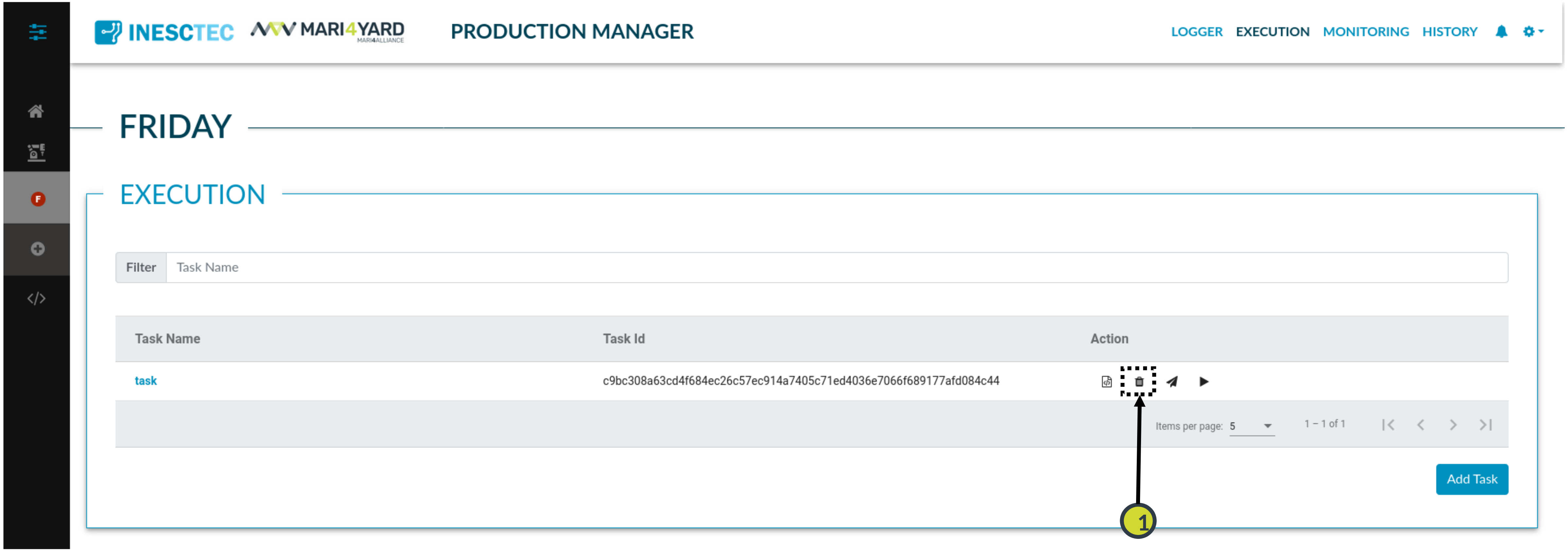
Click the Skill you want to inspect

3

Select desired Skill information (only most recent information is kept)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Production Manager – Delete Task



Click on the "Delete Task" button

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Thank you for your attention!

This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Robotic Grasping

PhD. João Pedro Souza
Researcher
INESC TEC
iiLab

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Index

Robotic Grasping 101

Robotic Grasping Hands-on

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

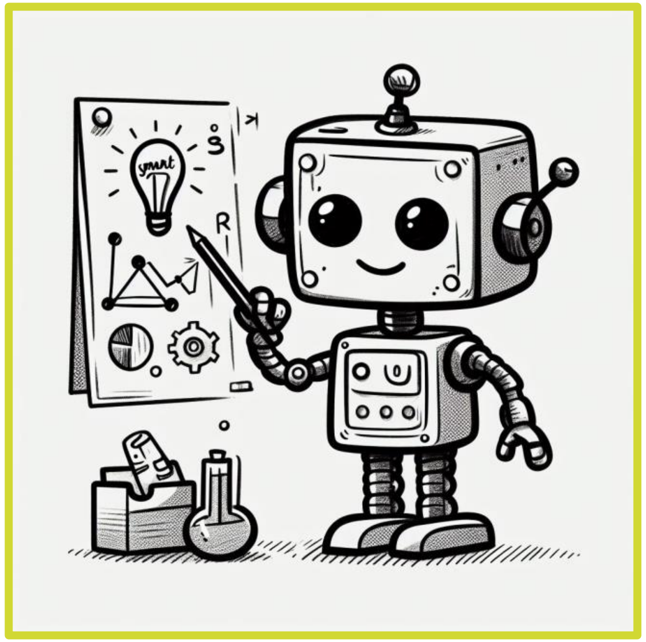
Learning Outcomes

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Learning Outcomes

- Define a grasping mission.
- List different sensing technologies.
- Describe object recognition strategies.
- List different gripper technologies.
- Categorize grasping techniques.
- Differentiate grasping approaches.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

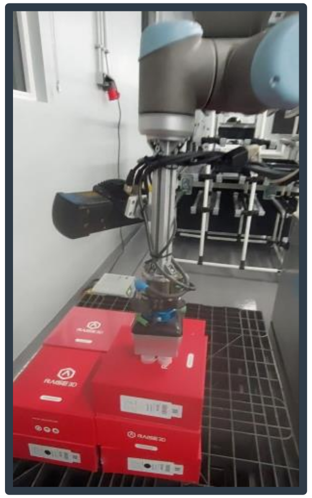
Why Robotic Grasping?

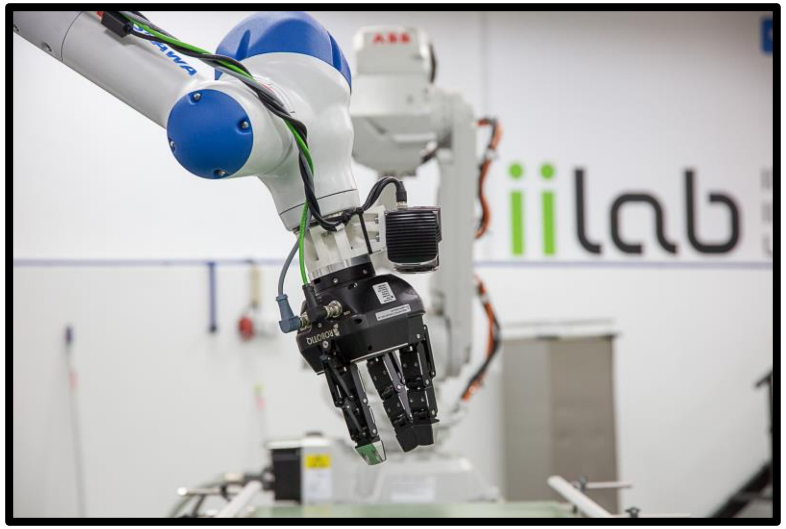
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Why robotic grasping?

- Robotic grasping is a fundamental and challenging skill in robotics.
- Several applications
  - Picking in the production line
  - Assembly
  - Machine Tending
  - Bin-picking
  - Interacting with humans in a collaborative manner



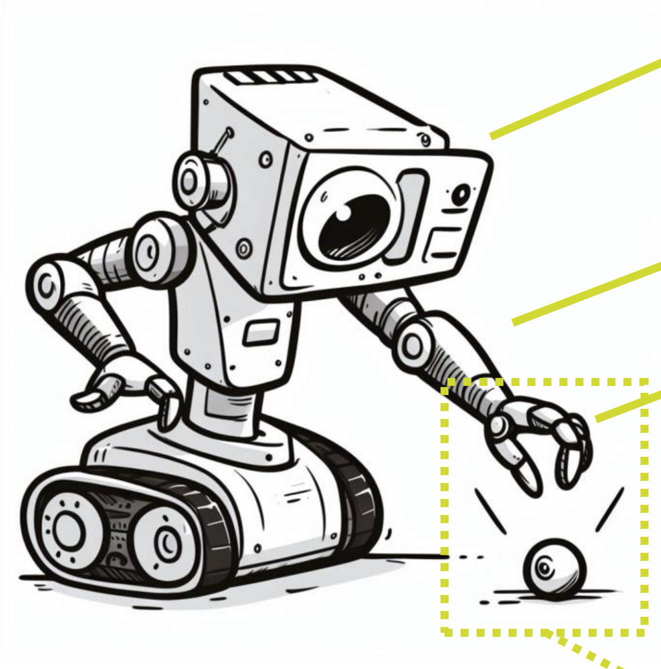


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Defining a grasping mission

We are focused on visual data!
- Image (RGB)
- PointCloud (3D)



Sensor
- The sensing (different technologies to generate data);
- The perception (give meaning to the sense data);
  - e.g. Image Segmentation and Object Recognition

Robotic Arm
Gripper
- The actuators (different technologies);
- The control and planning;
  - e.g. Grasping Strategies, Motion Planning and Robot Control

Active-Pair (Gripper + Workpiece)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

The prototype

- UR10 Robotic ARM
  - Payload 10kg
  - Reach 1,3m

- Photoneo Phoxi 3DCamera
  Accuracy < 0.300 mm

- RobotiQ 2f140
  Payload 2,5kg
  Finger distance 140mm

This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Robotic Grasping 101

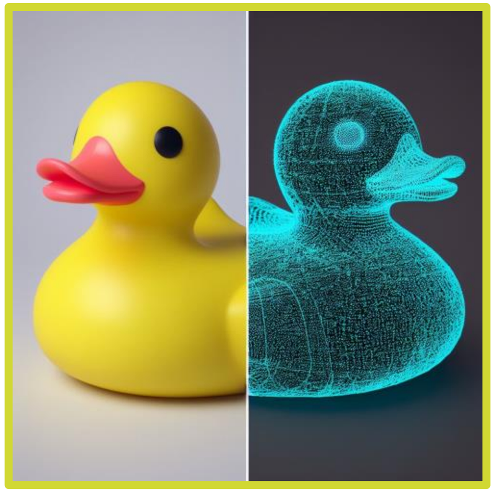
Visual Sensing Technologies

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Visual Sensing Technologies

Goal:
- Generate data for perception algorithms



Workpiece → Point Cloud

Two classes of image sensors:

1) Passive Sensor
2) Active Sensor

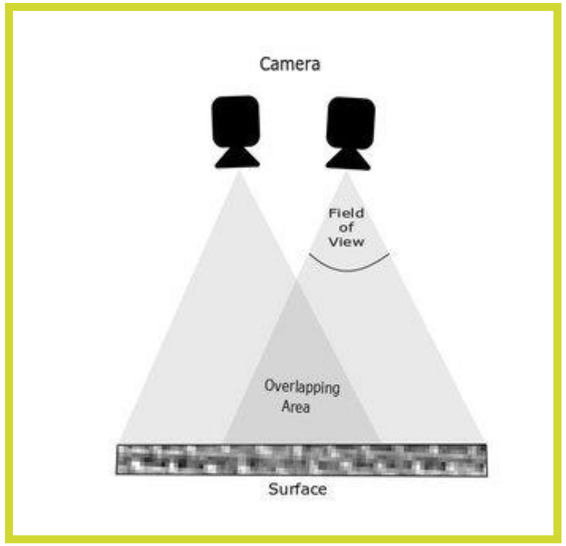
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

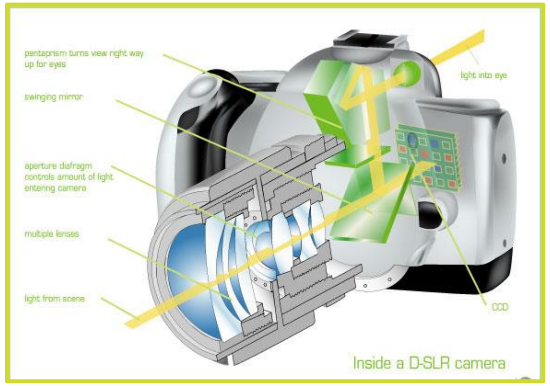
Sensing Technologies: Passive Sensor

- Does not interfere with the workpiece/environment
- Examples:
  - Monocular Cameras → 2D Image
  - Stereo Cameras → 3D Image

Stereo



Monocular



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

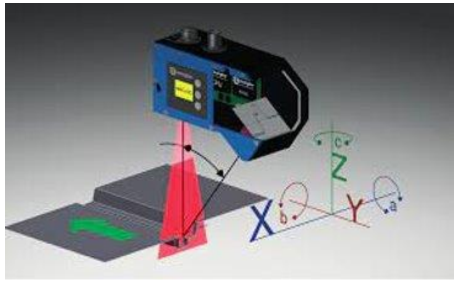
Sensing Technologies: Active Cameras

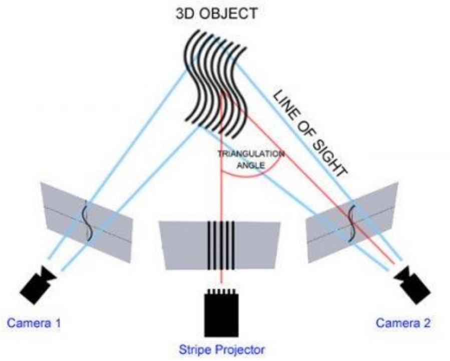
- Interfere with the workpiece/environment
- Examples:
    - Laser-point cameras
    - Laser-line cameras
    - Structured light cameras
    - LiDar

LiDar

Laser-Point

Laser-Line





Structured Light

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

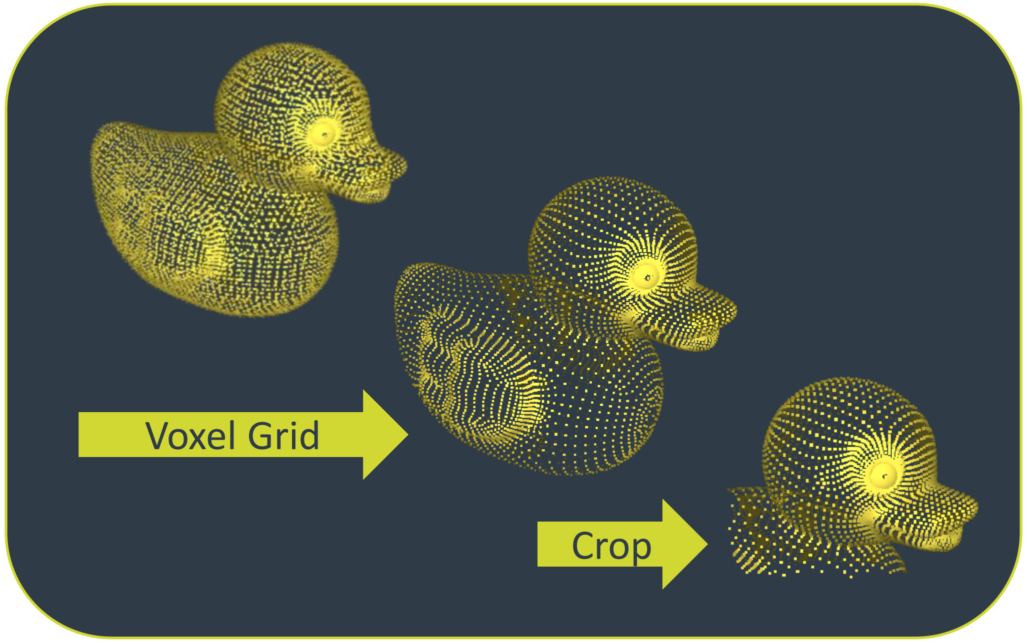
Robotic Grasping 101

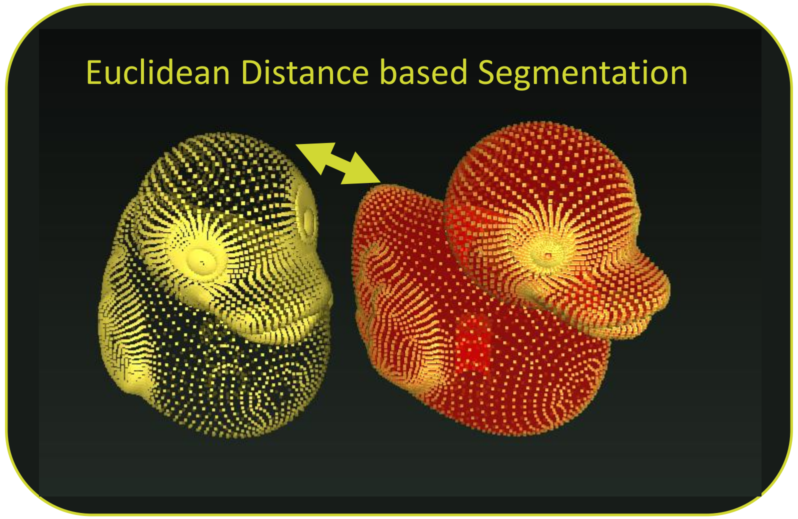
Perception

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101
Perception

- Data processing
  - Reduce data without losing information
  - Segment region of interest to speed processing
    - Cropbox
    - Voxel Grid
    - Region Growing
    - Euclidean Distance



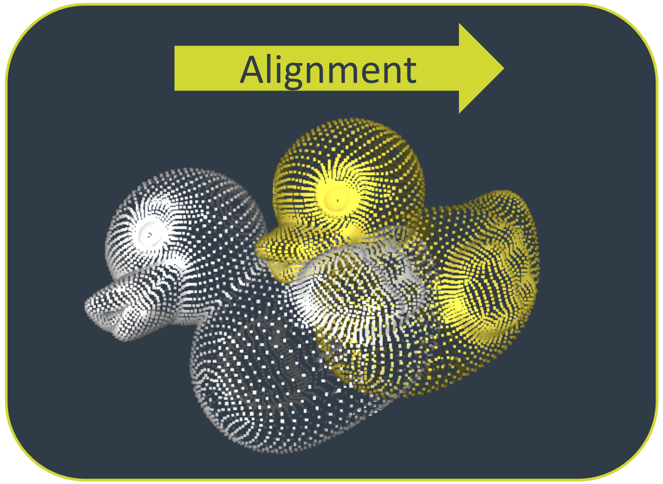


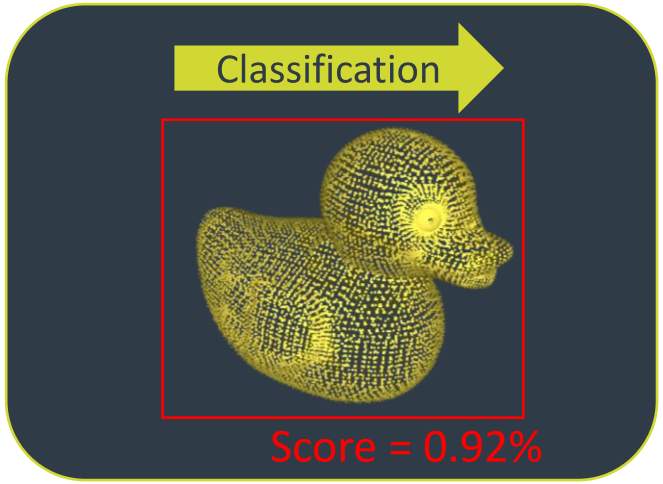
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Perception

- Object Recognition:
  - **Analytical:** face the problem as a matching or registration problem.
  - **Machine Learning:** face the problem as a classification problem.





- Registration problem:
  - ICP
  - Feature Matching (RANSAC)

- Classification problem:
  - CNNs
  - SVM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Gripper Technologies and Grasping Types

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Gripper Technologies and Grasping Types

Astrictive: binding force by field

Contigutive: contact prehension with the workpiece

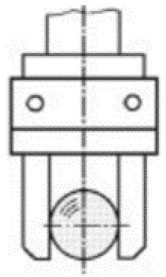
Ingressive: permeates the workpiece (intrusively or not)

Impactive: contact force with the workpiece (multi-finger)

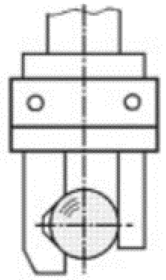
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

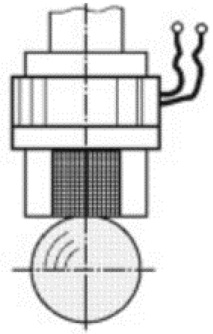
Gripper Technologies and Grasping Types



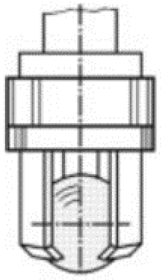
Force closure (Impactive)



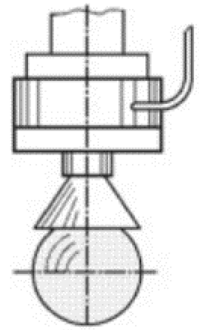
Force-form closure (Impactive)



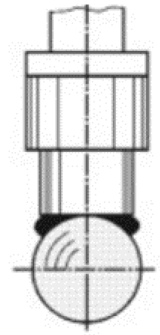
Magnetic closure (Astrictive)



Form closure (Impactive)



Pneumatic closure (Astrictive)



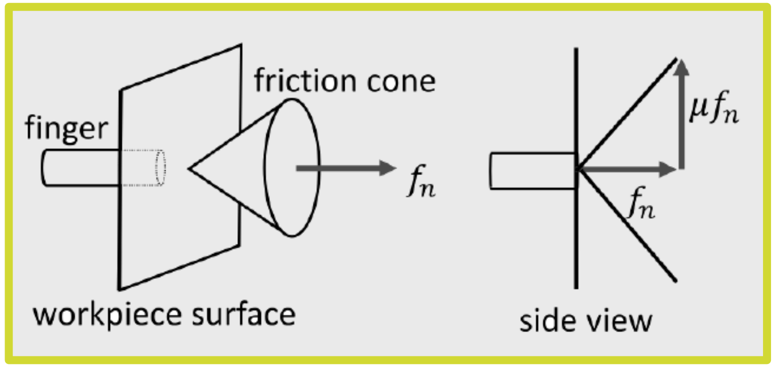
Adhesive closure (Contigutive)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Force closure formulation

- Model is important since:
  - Guarantee a stable grasping
  - Allow the robot movement control
  - Enable robot planning
- The basis relies on Coulomb Law of Friction
  - Considering the existence of friction contact
  - Considering no reactive torque



- **No slipping** occurs if the contact force is placed inside the represented cone defined by the normal force.

$$\mathbf{w_c} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}$$

$\boldsymbol{\tau}$ = 0

$$\mathbf{w} = \mathbf{f} \quad | \quad \mathbf{f} = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Force closure formulation

- After some math…

$$\mathbf{W_{c_i}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{w}_{c_i} \quad | \quad \mathbf{w}_{c_i} = \mathbf{f}_{c_i}, \quad \mathbf{f}_{c_i} \in FC_{c_i}$$

$$FC_{c_i} = \{\mathbf{f} \in \mathbb{R}^3 : \sqrt{f_x^2 + f_y^2} \leq \mu_t f_z,, f_z \geq 0\}.$$

$${}^{o}\mathbf{Tw}_{c_i} = \begin{bmatrix} {}^{o}\mathbf{R}_{c_i} & \mathbf{0} \\ {}^{o}\widehat{\mathbf{t}}_{c_i} {}^{o}\mathbf{R}_{c_i} & {}^{o}\mathbf{R}_{c_i} \end{bmatrix} \in \mathbb{R}^3$$

$${}^{o}\hat{\mathbf{t}}_{c_i} {}^{o}\mathbf{R}_{c_i} = \begin{bmatrix} 0 & -x & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \mathbf{R}_{c_i}$$

$$\mathbf{G}_i =^{o} \mathbf{Tw}'_{c_i} \mathbf{B}_{c_i}$$

$${}^{o}\mathbf{W} = [\mathbf{G}_1, \ldots, \mathbf{G}_N] \left[\mathbf{f}'_{c_1}, \ldots, \mathbf{f}'_{c_N}\right]' = \mathbf{GF}$$

where: $\mathbf{F} \in FC$ and $FC = FC_{c_1} \times \ldots \times FC_{C_N}$

$$\mathbf{f}_{ci} \approx \sum_{d=0}^{D} a_d \mathbf{s}_{ci_d}, a_d \geq 0 \quad | \quad \sum_{d=0}^{D} a_d = 1$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

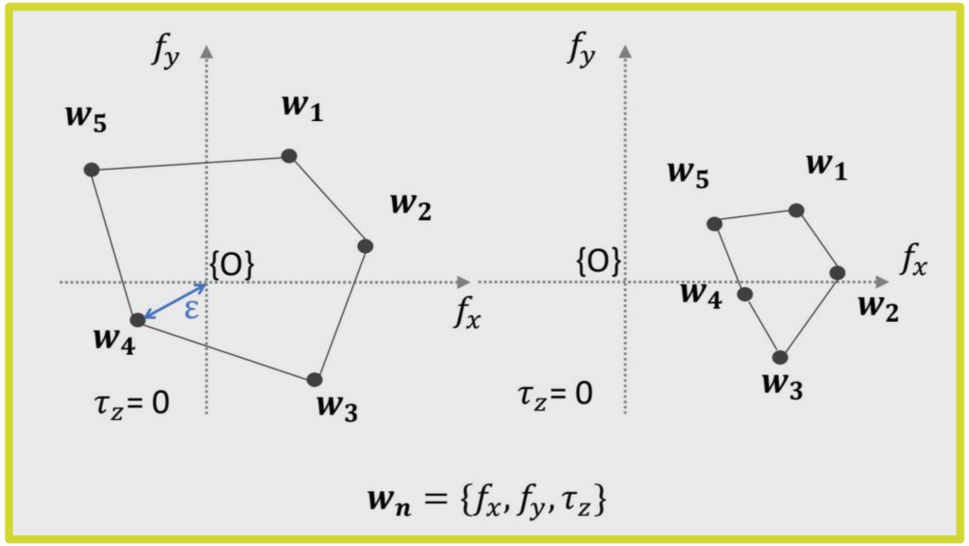
Robotic Grasping 101

Force closure formulation

- It is possible to define the wrench map!
  - Description of all contact forces associated with all fingers of an active-pair grasping.

$$^{o}\mathbf{W} = [\mathbf{G}_1, \ldots, \mathbf{G}_N]\,[\mathbf{f}'_{c_1}, \ldots, \mathbf{f}'_{c_N}]' = \mathbf{GF}$$

$$\mathbf{W}_p = [\mathbf{G}_1, \ldots, \mathbf{G}_N]\,[\mathbf{AS}'_{c1}, \ldots, \mathbf{AS}'_{cN}]' = \mathbf{GF_p}$$



- A stable grasp has its wrench map, including the origin.
- ε-value define the best one

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

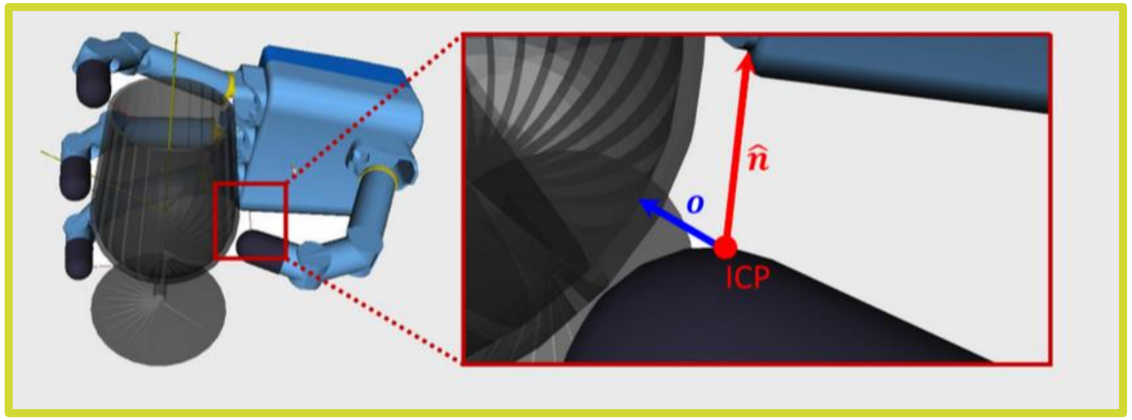
Force closure formulation

- Optimizing this wrench space can lead to the creation of a grasping dataset associated with the active pair.

$$Q = \sum_{i=1}^{N} (1 - \delta_i) \quad \text{with} \quad \delta_i = \frac{|\mathbf{o}_i|}{\alpha} + \left(1 - \frac{\hat{\mathbf{n}}_i \cdot \mathbf{o}_i}{|\mathbf{o}_i|}\right)$$

Simulated Annealing Optimization

- Therefore, simulation tools associated with the modelling are welcome!



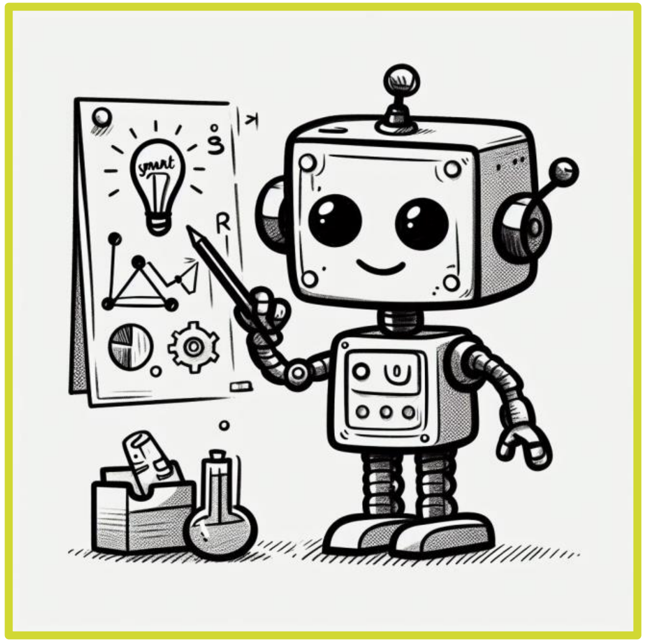
GraspIt! Simulator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping 101

Conclusion

- Define a grasping mission. ✓
- List different sensing technologies. ✓
- Describe object recognition strategies. ✓
- List different gripper technologies. ✓
- Categorize grasping techniques. ✓
- Differentiate grasping approaches. ✓



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping Hands-On

PhD. João Pedro Souza
Researcher
INESC TEC
iiLab

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping Hands-On

The perception and grasping system

This project has received funding from
the European Union’s Horizon 2020
research and innovation programme
under grant agreement No 101006798

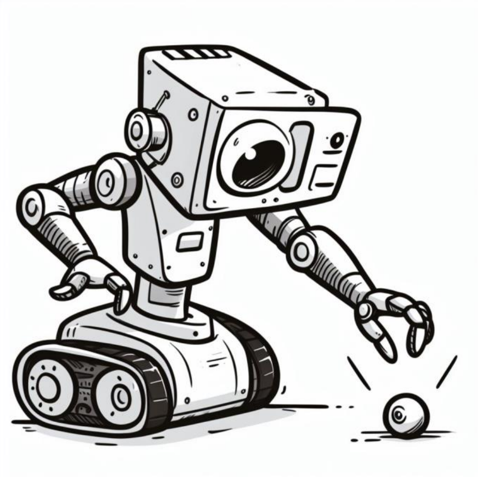
Robotic Grasping

The perception and grasping system

The developed system designed by INESC TEC is based on ROS package modules with Action Server implementation, called Skills. Each module is responsible for executing a specific task during the bin-picking procedure. The bin-picking could be divided into significant strategies, which led to the definition of two modules:

- **Object Recognition:** object localisation (i.e. estimate the position and orientation) given a point cloud image and a CAD reference model;

- **Grasping Estimation:** given a grasping dataset, select the best grasping posture given the mission's current environment restriction.



MARI4YARD
MARI4ALLIANCE

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

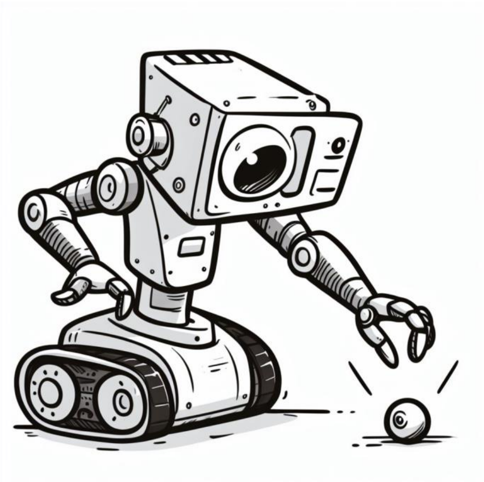
The perception and grasping system

A critical process in the object recognition step is segmentation (i.e. distinguishing the data of interest from the acquired data), which can be performed by analytical computer vision methods or artificial intelligence.

Since the INESC TEC focuses on simplicity and modularity, support tools orientated to IA are designed and listed below:

- **Image Segmentation:** IA-oriented capable of segmenting objects in a 2D image;

- **PointCloud Segmentation:** mapping the 2D segmented image to a 3D image, i.e. point cloud data;



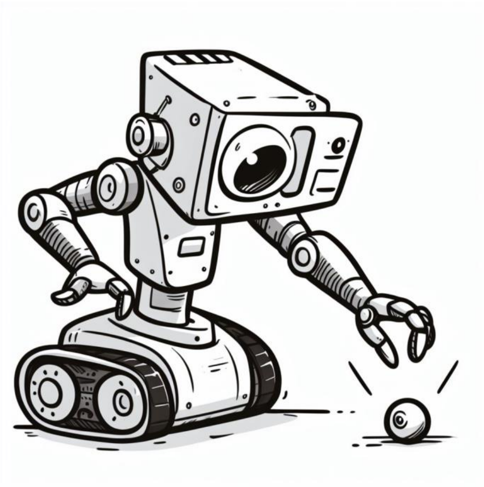
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

The perception and grasping system

Some described modules depend on datasets, such as Image Segmentation (to train the IA model) and Grasping Estimation (to generate the grasping postures). Other created designed modules facilitate the creation of these datasets. The course also includes the usage of these tools, which are:

- **Label Synthesis:** an automatic tool to generate large synthetic data to train IA models based on object CAD-model
- **Grasping Synthesis:** automatic generation of stable grasp postures given an active pair, i.e. object CAD-model and gripper type.



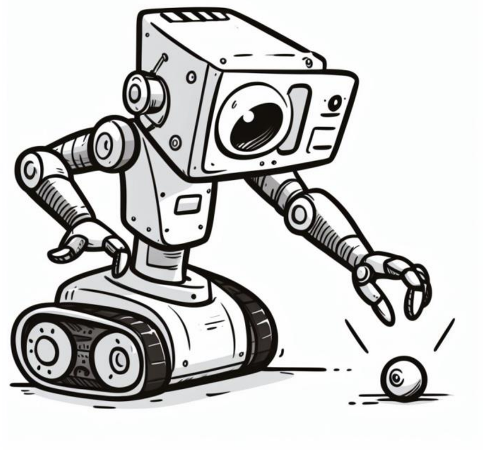
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

The perception and grasping system

- All modules respect:
  - the Textual Configurable Pipeline Paradigm
  - the ROS Action Server Paradigm

Sensing
↓
Image Segmentation
↓
Point Cloud Segmentation
↓
Object Recognition and Localisation
↓
Grasping Synthesis
↓
Grasping Estimation



This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Robotic Grasping Hands-On

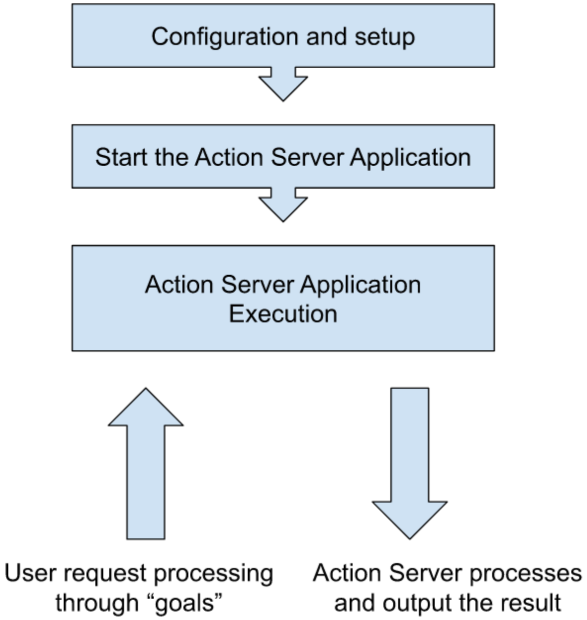
Paradigms

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

ROS and the Action Server Paradigm

- The Robot Operating System (ROS) is a set of software libraries and tools that help build robot applications. It provides hardware abstraction, device drivers, libraries, visualisers, message passing, package management, and more.
- One of the package programming styles in ROS is the Action Server. It provides a standardised interface for preemptable tasks, such as moving the base to a target location, performing a laser scan, and returning the resulting point cloud.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

The Textual Configurable Pipeline Paradigm

- Approach which enables users to construct a sequence of text blocks that outline the flow of pipeline processing.
- Each pipeline stage is defined as a "heuristic“ with unique characteristics described in a text block.

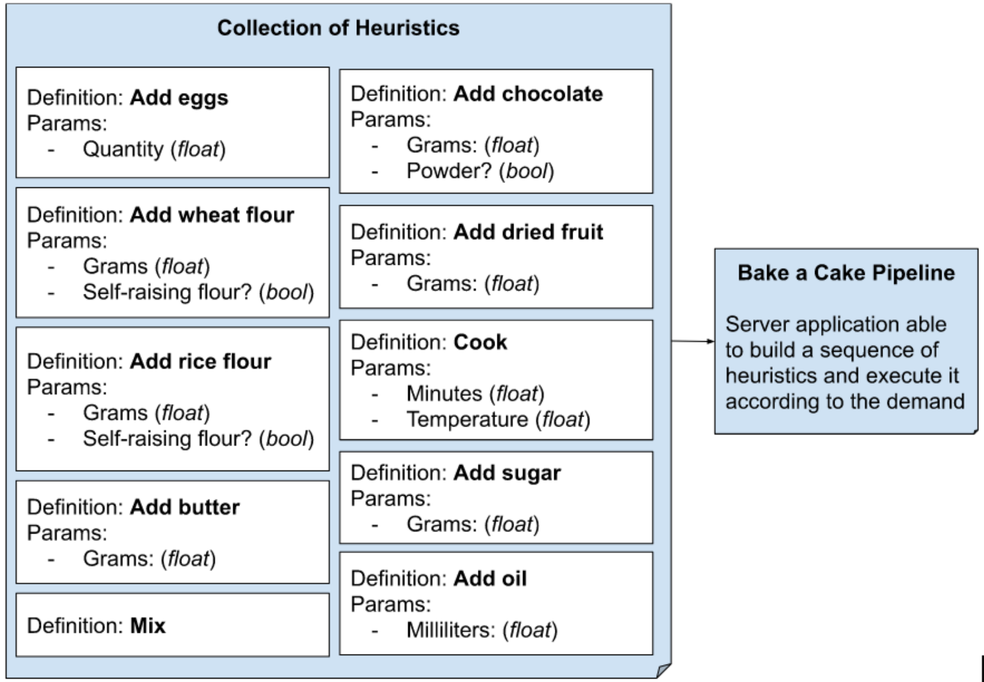
High Configurability ✓
High Flexibility ✓
High Modularity ✓
Easy to improve ✓
Easy to adjust ✓

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

The Textual Configurable Pipeline Paradigm

- The "bake a cake pipeline"

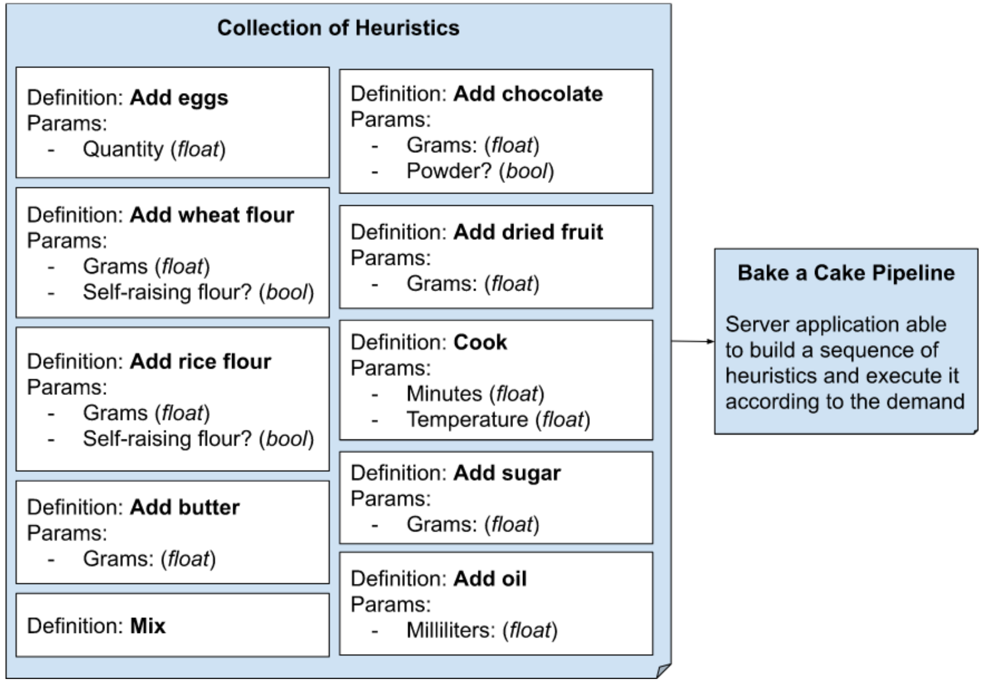


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

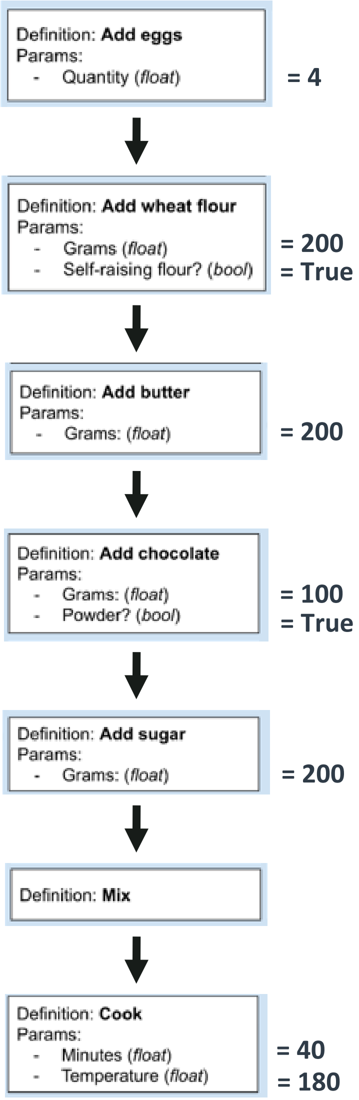
Robotic Grasping

The Textual Configurable Pipeline Paradigm

- The "bake a cake pipeline"



Guidelines to bake a chocolate cake?




```
pipeline:
    0_add_eggs:
        quantity: 4
    1_add_wheat_flour:
        grams: 200
        self_raising: true
    2_add_butter:
        grams: 200
    3_add_chocolate:
        grams: 100
    4_add_sugar:
        grams: 200
    5_mix:
    6_cook:
        minutes: 40
        temperature_in_celcius: 180
```

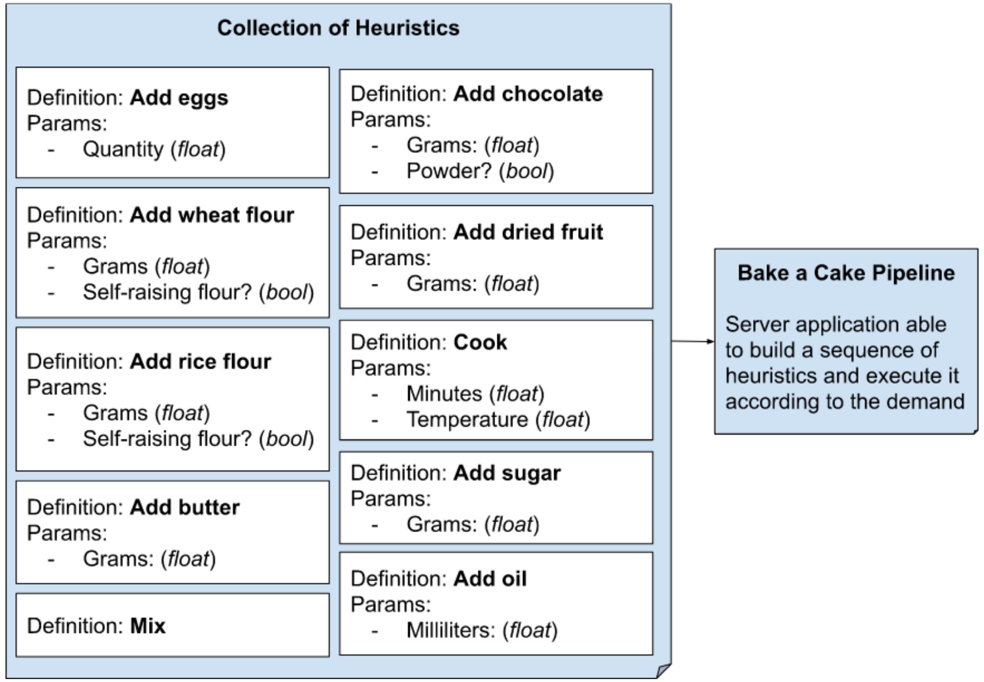
The textual configuration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

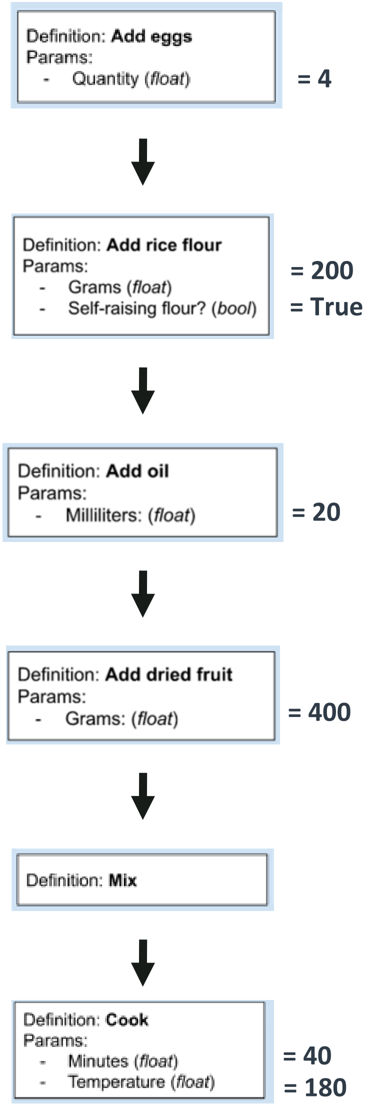
Robotic Grasping

The Textual Configurable Pipeline Paradigm

- The "bake a cake pipeline"



Guidelines to bake a fitness cake?




```yaml
pipeline:
    0_add_eggs:
        quantity: 4
    1_add_rice_flour:
        grams: 200
        self_raising: true
    2_add_oil:
        mililiters: 20
    3_add_dried_fruit:
        grasm: 400
    4_mix:
    5_cook:
        minutes: 40
        temperature_in_celcius: 180
```

The textual configuration

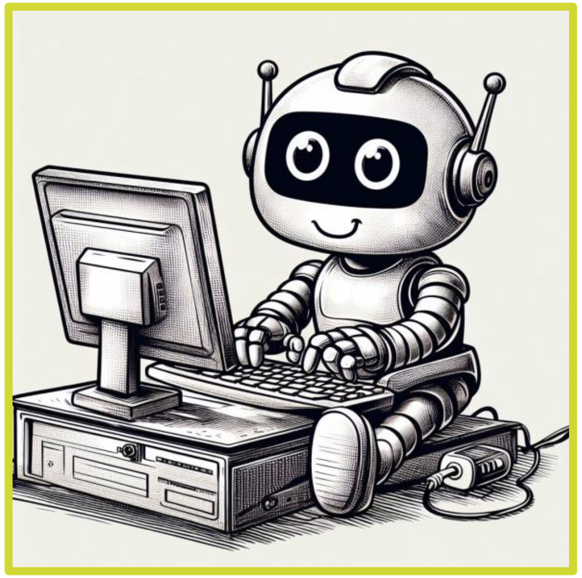
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

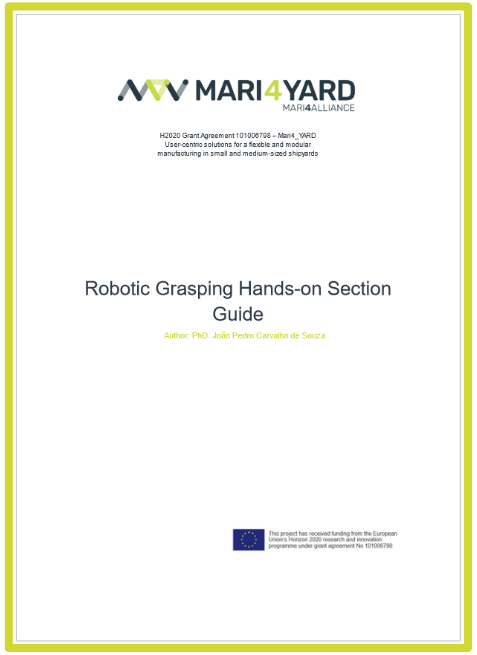
Robotic Grasping

Next Steps

MARI4YARD
MARI4ALLIANCE

HANDS-ON SECTION





H2020 Grant Agreement 101006798 – Mari4_YARD
User-centric solutions for a flexible and modular manufacturing in small and medium-sized shipyards

Robotic Grasping Hands-on Section Guide

Author: PhD. João Pedro Carvalho de Souza

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

*** Please refer to the additional document shared for further guidance on today's hands-on.

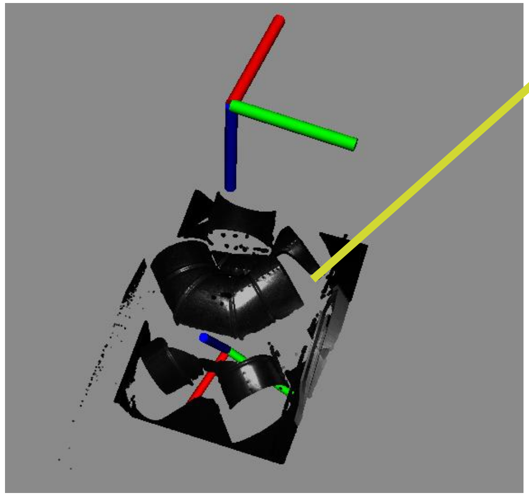
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

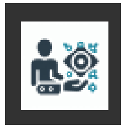
Robotic Grasping

Next Steps

Exercise 1) Object Recognition

Detect the tube in this 3D-image.





Object Recognition Icon

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

Object Recognition Heuristics:

- **Crop Box:** is a filter that allows users to filter all data within a given box. This method is useful for removing points that are outside of a region of interest.
- **Voxel Grid:** converts a 3D model into a three-dimensional grid of voxels, storing only a portion of the original points that reside within each voxel. This effectively reduces the number of points in a dataset and speed-up the processing time;
- **Random Sample:** used to reduce the number of points in a point cloud dataset. It randomly selects a subset of points from the original dataset;
- **Noise Removal:** identifies and removes outliers or noise in a point cloud. Common methods include statistical filtering, density-based clustering, and machine learning algorithms;
- **Normal Estimation:** calculates the surface orientation (normal) for each point in a point cloud. It is essential for many subsequent tasks which need any feature. Some sensors also provide this information, such as the Photoneo Sensor used in this course!
- **Plane Segmentation:** identifies and groups points that belong to the same plane in a point cloud. It is useful for extracting flat surfaces in 3D environments.
- **Euclidean Cluster Segmentation:** groups points that are close to each other in Euclidean space. It is useful for identifying distinct objects in a 3D scene.
- **Region Growing Segmentation:** groups points that are close neighbours and have similar properties, such as surface curvature or color. It is useful for segmenting complex objects that consist of multiple parts with different properties.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

Next Steps

Object Recognition Heuristics:

- Just activate or not

```
<arg name="use_voxel_grid" default="true" />
<arg name="use_random_sample" default="false" />
<arg name="use_noise_removal" default="true" />
<arg name="use_normal_estimation" default="false" />
<arg name="use_plane_segmentation" default="true" />
<arg name="use_euclidean_clustering_segmentation" default="true" />
<arg name="use_region_growing_segmentation" default="false" />
```

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

- The user needs to calibrate the configuration given a point cloud, but can this calibration generalise to an adversarial bin-picking situation?

- What could happen if the mobile robot platform positions deviate from the planned position?

- Or if the object is half-cut in the acquisition image?

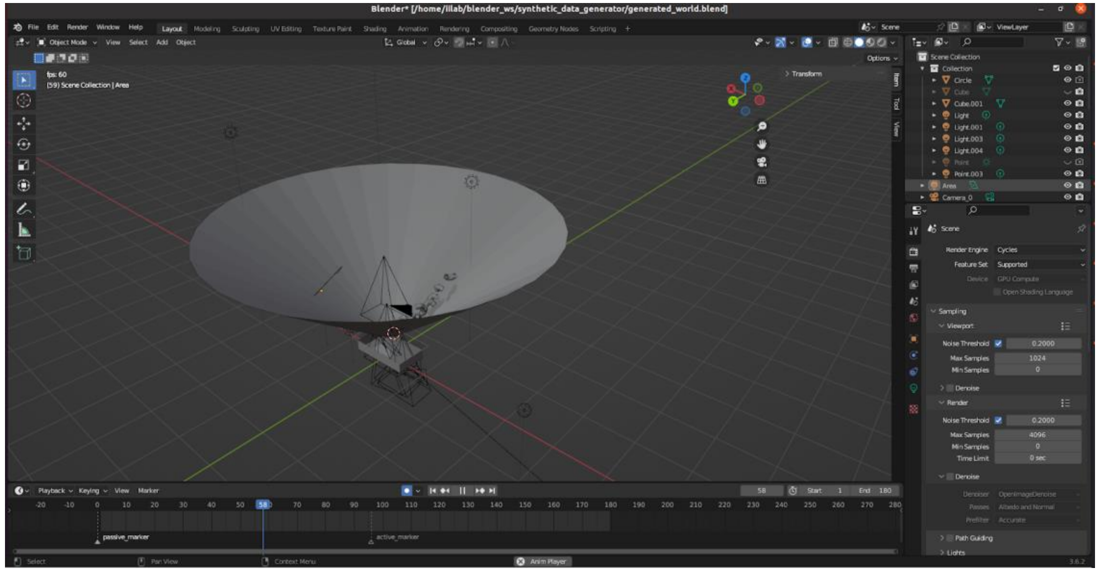
- Machine Learning Solution ←

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

Next Steps

- Machine Learning Solution:
  - Deep learning model using Convolutional Neural Network
  - Issue: Large dataset models.



Synthetic dataset generation

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

Exercise 2) Image segmentation

How to run the image segmentation pipeline?

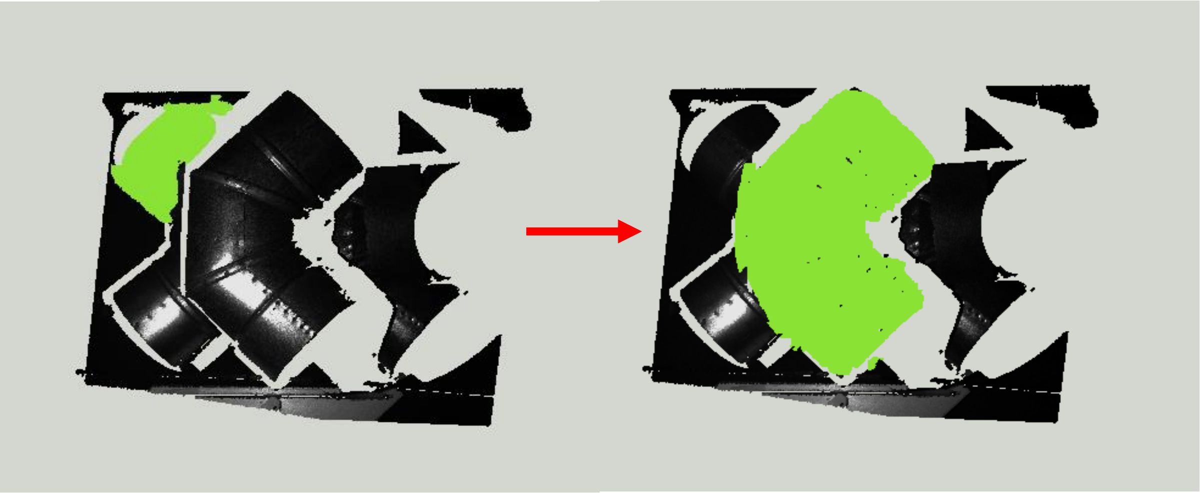
→ Image Segmentation Icon

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

Exercise 3 and 4) Point cloud segmentation

- How to run the 3D image segmentation pipeline? → Point Cloud Segmentation
- Is there anything wrong?
- How to improve it?



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

Next Steps

Point Cloud Segmentation Heuristics:

- **Area:** give priority to segmented point clouds with the biggest area in the image;
- **Depth:** give priority to near segmented point cloud;
- **Linear:** give priority to right most or/and left most segment point cloud in the image;

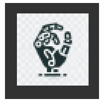
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

Next Steps

Grasping Synthesis

Verify the automatic generation by running…



→ Grasping Synthesis

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping

Next Steps

Point Cloud Segmentation Heuristics:

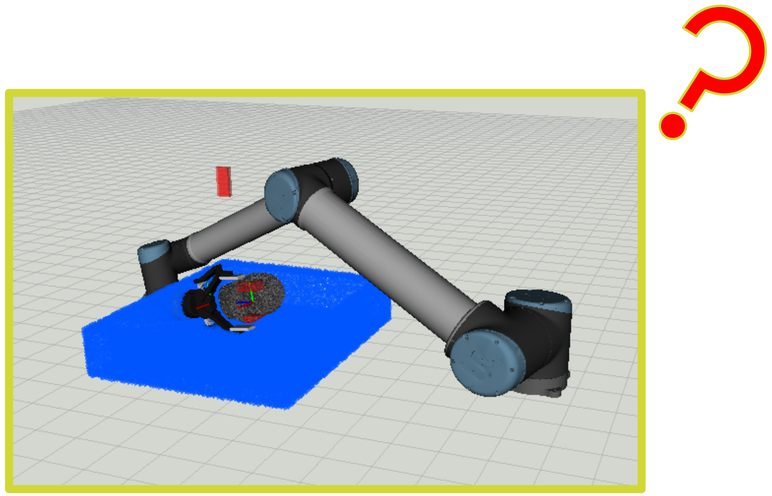
- **Area:** give priority to segmented point clouds with the biggest area in the image;
- **Depth:** give priority to near segmented point cloud;
- **Linear:** give priority to right most or/and left most segment point cloud in the image;

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

Exercise 5) Grasping Estimation

Provide the guidelines for a safe movement!



→ Grasping Estimation

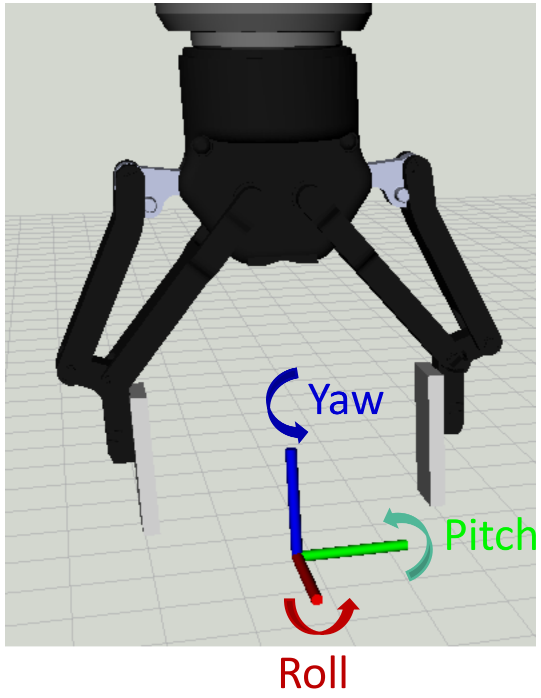
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

MARI4YARD
MARI4ALLIANCE

Grasping Estimation Heuristics:

- **Depth Distance:** this cost allows selecting candidates close to the depth from the current gripper pose.
- **Euclidean Distance:** this cost allows choosing near candidates from the current gripper pose considering all three dimensions.
- **Center of Gravity Distance**: select the postures near the object's centre of gravity
- **Roll, Pitch and Yaw Distances:** this scorer is a less effort angle displacement selector
- **Joint Space Filter:** In run-time, some grasping candidates can lead the robot to unfeasible kinematic configurations. Thus, aiming to avoid this, the Joint Space Filter heuristic calculates each candidate kinematic chain and discards the ones that exceed joint thresholds.
- **Workspace Filter:** The workspace filter is a method to discard candidates that exceed a spherical workspace threshold. This is useful to eliminate candidates with dangerous approach/lifting vectors.
- **Collision Filter:** The collision filter is a method that discards candidates that cause collision between the gripper's finger and the scene (or other objects).

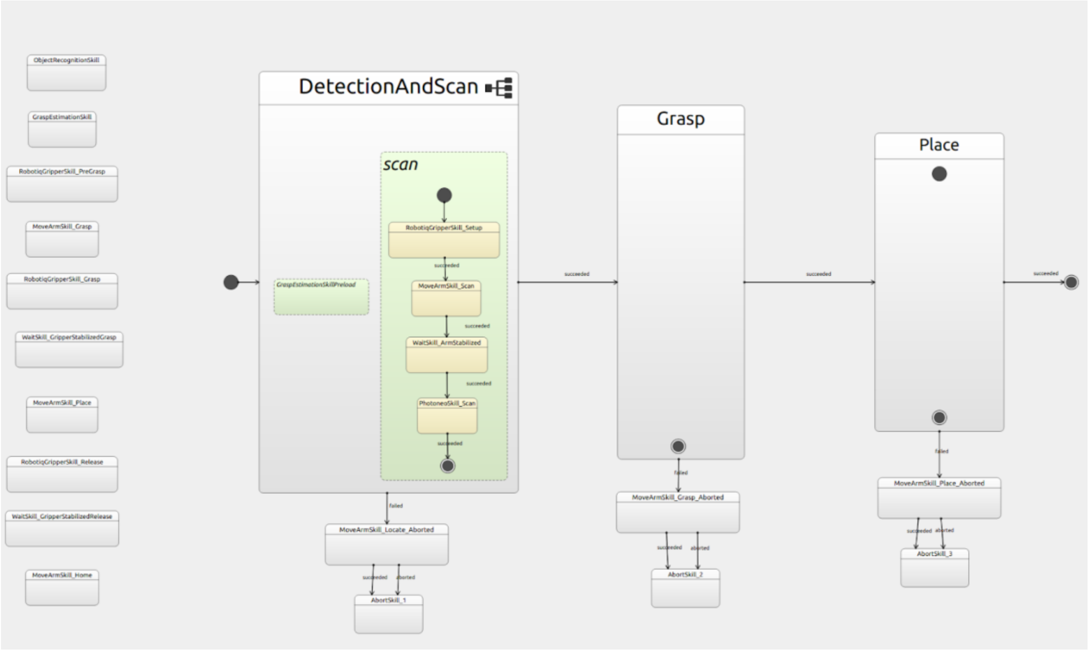


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Robotic Grasping
Next Steps

Exercise 6) Grasping Estimation

Build a complete mission.



This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Thank you for your attention!

João Souza | Researcher

joao.p.souza@inesctec.pt

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

H2020 Grant Agreement 101006798 – Mari4_YARD
User-centric solutions for a flexible and modular manufacturing in small and medium-sized shipyards

Robotic Grasping Hands-on Section Guide

Author: PhD. João Pedro Carvalho de Souza

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Summary

1. Technical Background

This section will present a summary of essential concepts which allow a complete understanding of the developed system in the context of Mari4Yard.

1.1. Linux Terminal

The Linux terminal is a text interface for your computer. They are often called shell, terminal, console, prompt, or other names. Users can enter commands to interact with the Linux operating system through it.

To run a program in the Linux terminal, first open the terminal. This can be done by pressing Ctrl + Alt + T on most Linux distributions or clicking the terminal icon in the application menu. After opening the terminal, type the name of the program and press Enter to run it.

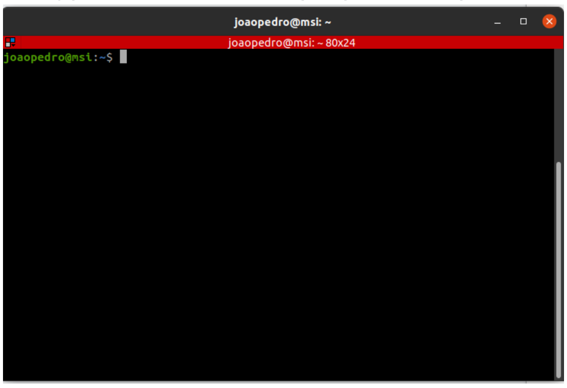


Fig 1. Linux Terminal.

In addition to running programs, the Linux terminal allows users to navigate through folders in the file system. To change to a specific directory, use the cd [foldername] command. To list files and folders in the current directory, use the ls command.

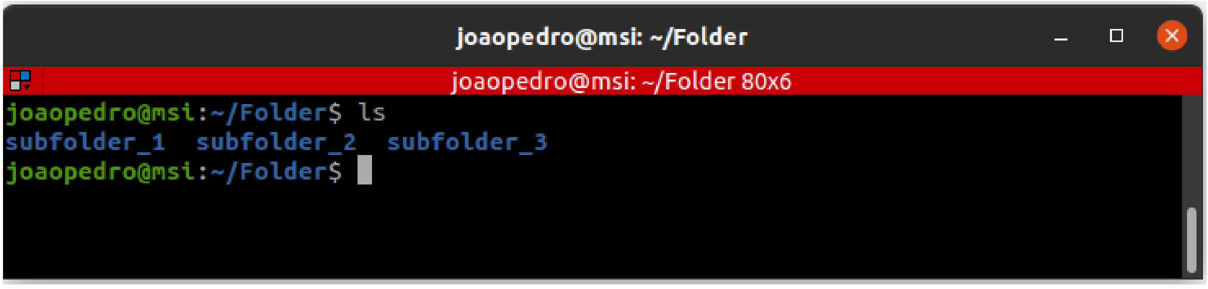


Fig 2. Listing command ls output.

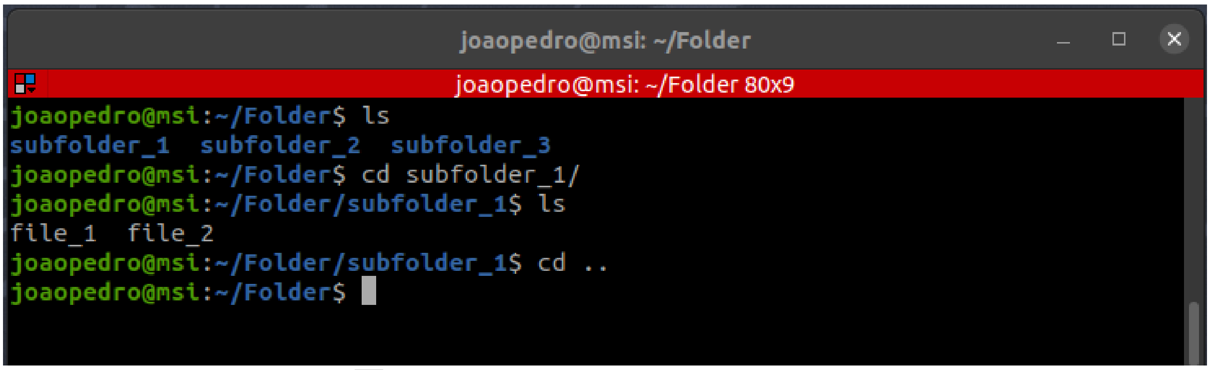


Fig 3. Changing directory by cd command. In the image, the cd command directs the user to "subfolder_1" (located inside the "Folder" directory). The cd .. command allows the user to return to the parent directory "Folder".

1.2. ROS and the Action Server Paradigm

The Robot Operating System (ROS) is a set of software libraries and tools that help build robot applications. It provides hardware abstraction, device drivers, libraries, visualisers, message passing, package management, and more. ROS is licensed under a BSD open-source license.

One of the package programming styles in ROS is the Action Server. It provides a standardised interface for preemptable tasks, such as moving the base to a target location, performing a laser scan, and returning the resulting point cloud. The Action Server and Action Client via the "ROS Action Protocol", built on ROS messages. The client and server provide a simple API for users to request goals (on the client side) or execute goals (on the server side) through function calls and callbacks.

It is also possible to configure the Action Server processing pipeline before the application execution by setting up text files with the ".yaml" extension.

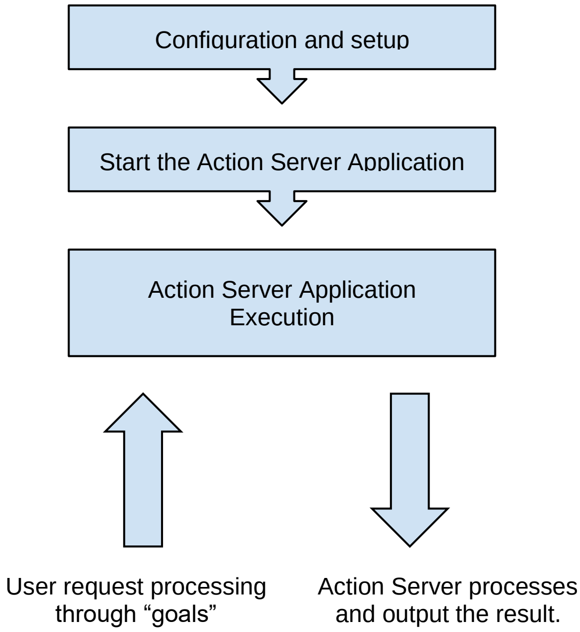


Fig 4. Action Server processing sequence.

1.3. The Textual Configurable Pipeline Paradigm

The modules of the system are designed following the paradigm of a configurable textual pipeline. This approach enables users to construct a sequence of text blocks that outline the flow of pipeline processing. Each pipeline stage is defined as a "heuristic", with its unique characteristics described in a text block. By grouping and ordering a collection of heuristics, users can create various strategies to meet their needs.

Consider the example of a "bake a cake pipeline". A heuristic might be a cooking step, such as adding 300 grams of wheat flour with yeast, or adding two eggs, or mixing the ingredients (Fig 5.). The process could be an ordered sequence of these steps. However, just like in real life, if the dough turns out too wet, we could add another "add wheat flour" heuristic with a smaller amount, say 50 grams, to stiffen the dough without starting over.

Furthermore, the same flow strategy, or in other words, the same server application, can be used to bake a different type of cake, such as a fitness cake, instead of a chocolate one, as represented in Fig 6. Therefore, this approach provides highly configurable pipeline strategies to cater to robotic requirements.

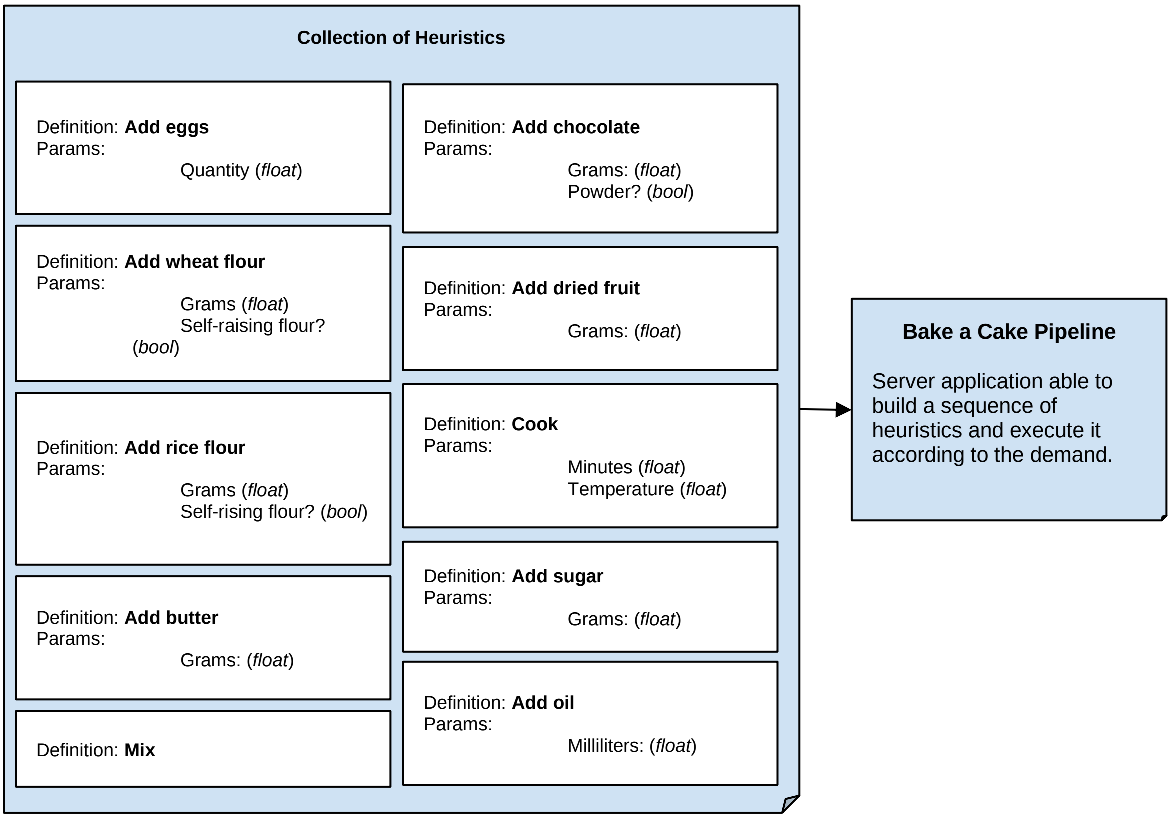


Fig 5. Collection of implemented heuristic to the “cake a bake” pipeline.

```yaml
pipeline:
    0_add_eggs:
        quantity: 4
    1_add_wheat_flour:
        grams: 200
        self_raising: true
    2_add_butter:
        grams: 200
    3_add_chocolate:
        grams: 100
    4_add_sugar:
        grams: 200
    5_mix:
    6_cook:
        minutes: 40
        temperature_in_celcius: 180
```

```yaml
pipeline:
    0_add_eggs:
        quantity: 4
    1_add_rice_flour:
        grams: 200
        self_raising: true
    2_add_oil:
        mililiters: 20
    3_add_dried_fruit:
        grasm: 400
    4_mix:
    5_cook:
        minutes: 40
        temperature_in_celcius: 180
```

Fig 6. Pipeline flow configuration in textual. (left) A chocolate cake pipeline; (right) A fitness cake pipeline. Both strategies use the same application with customized heuristic flows.

2. System Structure

The developed system designed by INESC TEC is based on ROS package modules with Action Server implementation, called Skills. Each module is responsible for executing a specific task during the bin-picking procedure. The bin-picking could be divided into significant strategies, which led to the definition of two modules:

1. **Object Recognition:** object localisation (i.e. estimate the position and orientation) given a point cloud image and a CAD reference model;
2. **Grasping Estimation:** given a grasping dataset, select the best grasping posture given the mission's current environment restriction.

A critical process in the object recognition step is segmentation (i.e. distinguishing the data of interest from the acquired data), which can be performed by analytical computer vision methods or artificial intelligence. The current course shows that the analytical strategies could solve a wide range of related issues in contrast to the time-spend configuration step. Another negative point is the reduced generalisation capability. This generalisation is also important in dense clutter bin-picking problems, which is more suitable to random issues. Since the INESC TEC focuses on simplicity and modularity, support tools orientated to IA are designed and listed below:

3. **Image Segmentation:** IA-oriented capable of segmenting objects in a 2D image;
4. **Point Cloud Segmentation:** mapping the 2D segmented image to a 3D image, i.e. point cloud data;

Some described modules depend on datasets, such as Image Segmentation (to train the IA model) and Grasping Estimation (to generate the grasping postures). Other created designed modules facilitate the creation of these datasets. The course also includes the usage of these tools, which are:

1. **Label Synthesis:** an automatic tool to generate large synthetic data to train IA models based on object CAD-model
2. **Grasping Synthesis:** automatic generation of stable grasp postures given an active pair, i.e. object CAD-model and gripper type.

The development of the modules has the paradigm of pipelining, discussed in Section 1.3. The user can set the module process by selecting and setting different process methods in a pipeline sequence. The pipeline is defined by a configuration file with extension .yaml and is configured before running the module, i.e. in an offline phase. This strategy is aligned with the modularity and flexibility paradigms, allowing the system deployment in different scenarios.

The following sections show how to operate each listed module, followed by a practical exercise.

2.1. Object Recognition

The designed object recognition solution allows different configurations and pipeline structures. Therefore, the current section will cover configuring the system and aligning it with the robot manipulation task.

The first thing to consider is the sensor data. Basically, the sensing data is a set of 3D points, which are called point cloud data. Together with this RGB data is also provided. For the current section use-case, a pre-capture point cloud will be used.

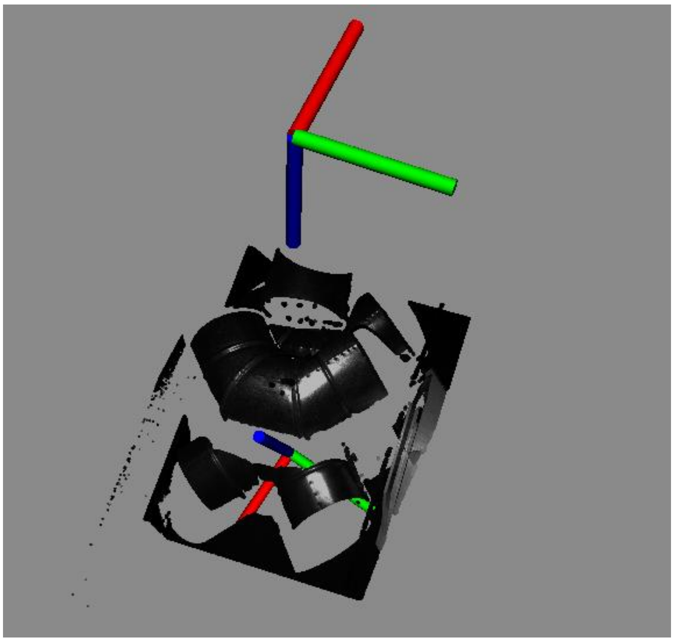


Fig 7. A point cloud image.

The object recognition system can detect objects by aligning features over an environment given an object CAD model, i.e. the algorithm tries to minimize the distance between the features of the sensing data and the 3D object model. However, to achieve this stage of processing, the input point cloud should be pre-processed with different computer vision techniques. The list below shows some of these techniques supported by the pipeline, i.e. the following heuristics are supported:

Crop Box: is a filter that allows users to filter all data within a given box. This method is helpful in removing points outside a region of interest.

Voxel Grid: converts a 3D model into a three-dimensional grid of voxels, storing only a portion of the original points within each voxel. This effectively reduces the number of points in a dataset and speeds up the processing time.

Random Sample: used to reduce the number of points in a point cloud dataset. It randomly selects a subset of points from the original dataset.

Noise Removal: identifies and removes outliers or noise in a point cloud. Common methods include statistical filtering, density-based clustering, and machine learning algorithms.

Normal Estimation: calculates the surface orientation (normal) for each point in a point cloud. It is essential for many subsequent tasks which need any feature. Some sensors provide this information, such as the Photoneo Sensor used in this course!

Plane Segmentation: identifies and groups points that belong to the same plane in a point cloud. It is helpful in extracting flat surfaces in 3D environments.

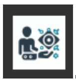
Euclidean Cluster Segmentation: groups points that are close to each other in Euclidean space. It is helpful in identifying distinct objects in a 3D scene.

Region Growing Segmentation: groups points that are close neighbors and have similar properties, such as surface curvature or colour. It is useful for segmenting complex objects with multiple parts with different properties.

Exercise 1:

The reader is encouraged to build an object recognition solution based on the listed techniques. Given a raw point cloud, the core idea is pre-processing it to facilitate the implemented localisation algorithm.

To do so, open the file “config_or” located on Desktop. This file allows you to activate the heuristics by setting “true” or deactivate by setting “false”. The " heuristics " directory in Desktop allows you to configure each heuristic individually. **Any doubt, call the professor. Let’s go hands-on!**

To run the object recognition, click its icon .

A window will pop up the RViz interface (Fig 8.), and in the right column, specific visualisations can be selected, e.g., the text box relative to Sensor Data to visualise the raw point cloud. However, the other topics will only be loaded after the pipeline execution; to do so, the user must request the processing by sending an action goal. Therefore, in the RQT window (Fig 9.) select the object recognition goal, press the right mouse button and press “Publish selected once”. Now the pipeline will run, and all topics can be verified, such as Voxel Grid, Filtered Cloud, Cropbox and so on…

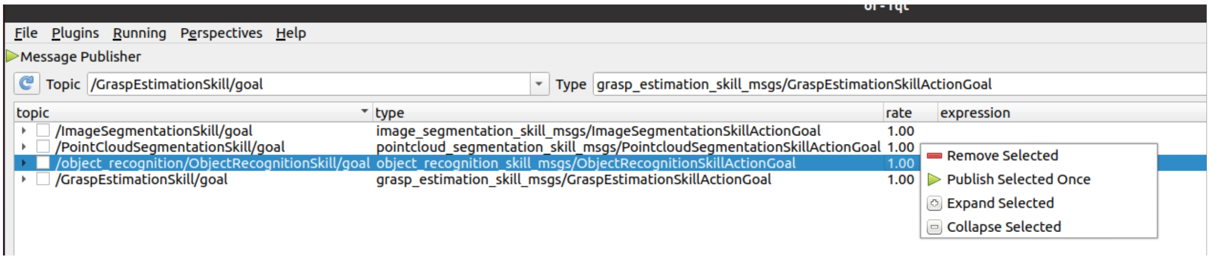


Fig 8. The RQT interface provides an Action Goal interface.

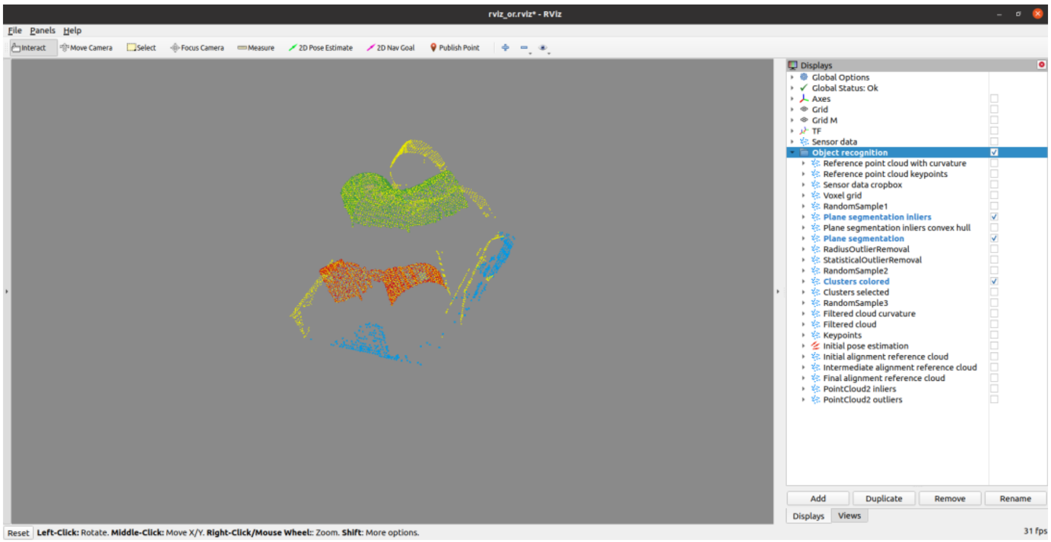


Fig 9 . The RViz interface provides a visualization object recognition tool.

As can be seen in the last exercise, the user needs to calibrate the configuration given a point cloud, but is this calibration able to generalize to an adversarial bin-picking situation? What could happen if the mobile robot platform positions deviate from the planned position? Or if the object is half-cut in the acquisition image?

Trying to solve these problems, an automatic segmentation system was designed and it will be discussed in the next sections.

After the end, close all the windows.

2.2. Label Synthesis

This module automatically creates a diverse synthetic dataset used in IA model training. The image segmentation module will use this IA module, allowing the robot to segment the object of interest in the acquired scenario image. After training, the AI module can be reused. In other words, this procedure is only necessary when configuring new objects.

To create this dataset, the object CAD model is necessary. The Blender generates the synthetic world, and the user needs to launch the system with the object to be used. This synthetic world creates different objects' bin-picking configurations and builds several images of these scenes.

The instructor will present this procedure since it demands graphical computer resources.

Wait for the process to finish. In Blender, it is possible to check the simulation running by selecting the " Layout " option in the menu bar.

Fig 10. Synthetic world in blender.

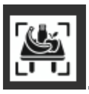
After the simulation conclusion, the system will start the labeling, which is the action of annotation, i.e. indicating where the object of interest is in the image. This process is time-consuming to be performed by hand. Therefore, the created label synthesis module supports this process by automatically building these annotations. The results can be checked in the folder “synthetic_data_generator>generated_data”.

2.3. Image Segmentation

The AI model could be trained once the Label Synthesis solution creates the generated dataset. The training procedure will not be addressed in the current course since the training demands time and good processing. The exported trained model, a Mask-RCNN model, will be used on the Image Segmentation, and it is provided by the file “knee_tube_weight.h5” located in “mari4yard_course>config>image_segmentation”.

The Mask-RCNN is a convolutional neural network (CNN) that is state-of-the-art in image segmentation. It is a deep neural network variant that detects objects in an image and generates a high-quality segmentation mask for each instance. This model was trained using a PC with an Intel i9 processor, 64Gb DDR4 RAM and a GeForce 4090RTX 24Gb of VRAM.

Exercice 2

To test the trained models, click on its icon .

In RQT, send the Image segmentation goal by clicking with the right button mouse and selecting” Publish Selected Once” (Fig 11.).

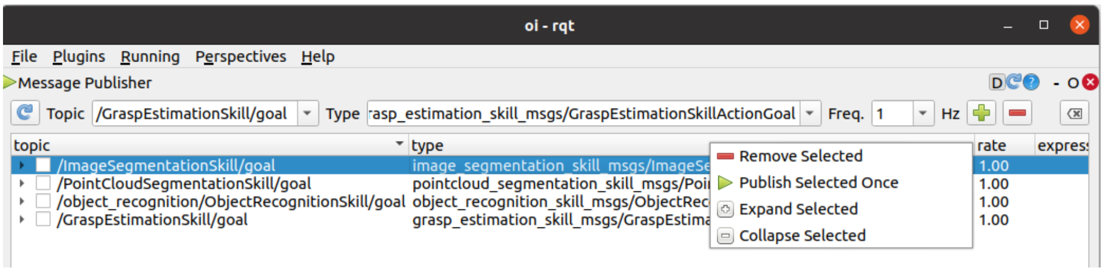


Fig 11. Sending goal request to Image Segmentation pipeline.

The result window shows 2D images with segmented objects.
After the end, close all the windows.

2.4. Point Cloud Segmentation

Since we are dealing with 3D information, a module to map the 2D segmentation image to the point cloud is created. Called Point Cloud Segmentation, this module creates a correspondence between segmented 2D pixels into point clouds and separates the region of interest. Therefore, this step can substitute the manual procedure discussed in Section 2.1.

To perform a segmentation test, do click its icon .

1. In RQT, select the Image Segmentation Goal and send a request; right click on the topic “/ImageSegmentationSkill/goal” and send the goal;
2. Wait for the image processing run.
3. In RQT, Select the Point Cloud Segmentation Goal and send a request with the right click on the topic “/PointCloudSegmentationSkill/goal” and send the goal;
4. Wait a little bit.
5. Verify the output in the RViz window;

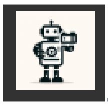
Since this is an automatic tool, it would be interesting to provide guidelines to the robot to select the best point cloud according to the demand. Therefore, the module was designed in a pipeline format, and the following selection heuristics can be defined:

Area: give priority to segmented point clouds with the biggest area in the image;
Depth: give priority to near-segmented point cloud;
Linear: give priority to the right-most or/and left-most segment point cloud in the image;

Exercise 3
Build the pipeline to select the middle knee tube by selecting the adequate sequence of heuristics (such as area, depth and/or linear). To do so, close the point cloud segmentation application (if opened) and edit the file “config_pcs” in Desktop. Save it

and relaunch it by clicking its icon . Re-execute the Goal requests such as in the last steps list.

2.5. The perception system

To verify the complete perception solution, i.e. Point Cloud Segmentation and Object Recognition, click the icon .

Exercise 4:
Could you infer what should be the sequence of goal calls to perform the overall process?

After the end, close all the windows.

2.5. Grasping Synthesis

To generate grasping candidates an automatic tool was designed based on the ROS system and "GraspIt!" simulator. The "GraspIt!" is a public domain simulator software focusing on robotic grasping physics simulation.

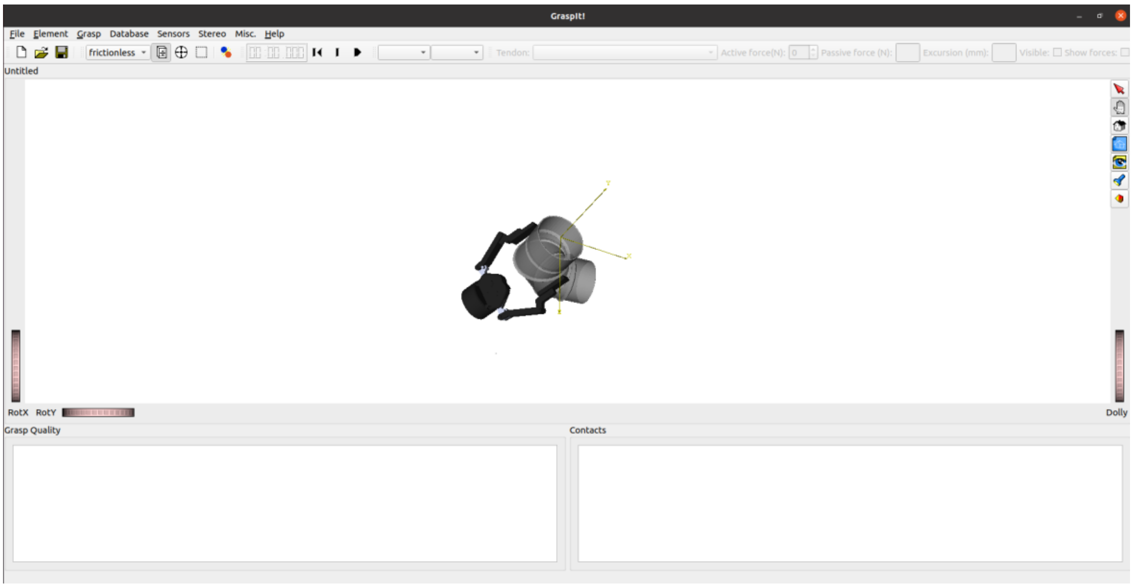


Fig 12. The "GraspIt!" simulator.

The proposed system is able to interact with the "GraspIt!" simulator, configure it, load the correct parameter and scene objects followed by the gripper model. To perform the automatic generation an optimization algorithm is used, called Simulated Annealing (SANN). This strategy allows us to find stable contact points between the gripper finger and the object of interest automatically. The grasp stability is analyzed and all feasible grasps are stored in memory and, in the end, the system exports the dataset in the format created for the designed grasping system.

Post-processing heuristics are also implemented allowing that, before generating the dataset, the system is able to expand the dataset, filter undesirable grasping configuration and show the generated solutions.

Select the file “config_gs” in the Desktop area to configure the system.

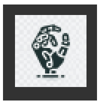
The opened file follows the premise of pipelining processing. The file is divided into modules with an order prefix number. Each module has its configuration parameters structured in an indented tree.

The core idea is the code running the modules in order to achieve its object. In the current case, the SANN automatic algorithm is executed followed by the distance filter and the tool center point angle twin.

SANN: allow the finding of grasping configurations given the active-pair (gripper and object model). Regarding the parameters, we focus in the current course on only defining the active-pair model name. A model library was created, and for our practical exercise, the “robotiq_2f_140_outer_finger/robotiq_2f_140.xml” is defined by the “knee_tube.ply” object. This gripper is the model used in current robot solutions and has two fingers. An important parameter is the interaction, which defines how many iterations will try to find the grasping solutions.

Distance filter: after generating all grasping postures (aka candidates) by SANN algorithm, this filter will remove candidates near each other. To define what is “near”, some threshold values are defined in angle and distance.

Tool center point angle twin: since the gripper has a symmetry (i.e. it is possible to pick the object rotate 180º given a candidate) this heuristic allows to expand the dataset. Don’t forget we are trying to create diversity thus the robot could have decisions options while operating.

Therefore, to run the pipeline click its icon  .

OBS: with the simulation running, in the GraspIt! screen, click on the “Eye” button in the right bar to adjust the simulation visualisation scale. Now, it is possible to see the estimation being done.

Wait for the process to finish. Go to the terminal, type ctrl+c and save the exported the grasping dataset for later use.

To verify the resulting dataset, click on grasping viewer icon . Type ENTER in the terminal to navigate through it.

After the end, close all the windows.

2.6. Grasping Estimation

Once the dataset is built and the perception configured, the next step is defining the grasping guidelines the robot will rely on. Given the environment's limitations, these guidelines are essential since the robot should know which grasping posture is more appropriate. Therefore, the grasping estimation was created to allow the user to describe these guidelines in a format of heuristic pipelines.
The supported heuristics are described below:

Depth Distance: The Depth distance scorer is a method to set the grasping candidate cost value according to the depth distance between the TCP reference frame and the candidate.
Therefore, this cost allows selecting candidates close to the depth from the current gripper pose.

Euclidean Distance: The Euclidean distance scorer sets the grasping candidate cost value according to the Euclidean distance between the TCP reference frame and the candidate. This cost allows choosing near candidates from the current gripper pose considering all three dimensions.

Center of Gravity Distance: The Center of Gravity (COG) distance scorer is a method to set the grasping candidate cost value according to the Euclidean distance between the candidate and the object's COG reference frame. This cost is important in high-dimensional objects where grasping poses can cause torques over the centre of gravity and affect the equilibrium in grasping movements.

Roll, Pitch and Yaw Distances: The angle distance scorer is a less effort angle displacement selector. Since the "Grasping Selection" pipeline core relies on configurability, the angles and distance heuristics are developed independently once the application considers only a single rotation axis. Roll in X-axis, Pitch in Y-axis, Yaw in Z-axis.

Joint Space Filter: Some grasping candidates can lead the robot to unfeasible kinematic configurations in run-time. Thus, aiming to avoid this, the Joint Space Filter heuristic calculates each candidate kinematic chain and discards the ones that exceed joint thresholds. Besides each candidate, the approach pose to grasp is also evaluated.

Workspace Filter: The workspace filter is a method to discard candidates that exceed a spherical workspace threshold. This is useful to eliminate candidates with dangerous approach/lifting vectors. For instance, if the center of a box is defined as the sphere's origin, it is possible to avoid candidates that generates approach/lifting vectors that collide with the box border.

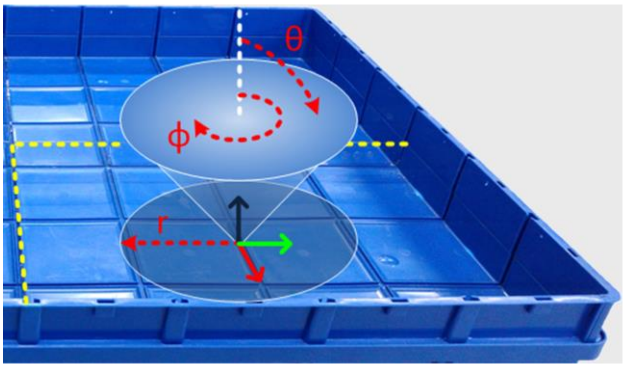


Fig 13. Workspace filter illustration.

Collision Filter: The collision filter is a method that discards candidates that cause collision between the gripper's finger and the scene (or other objects). The fingers trajectory is considered, i.e., the trajectory from open pose to close gripper's finger. The point clouds of the scene must be provided, and the collision shape volume must be defined.

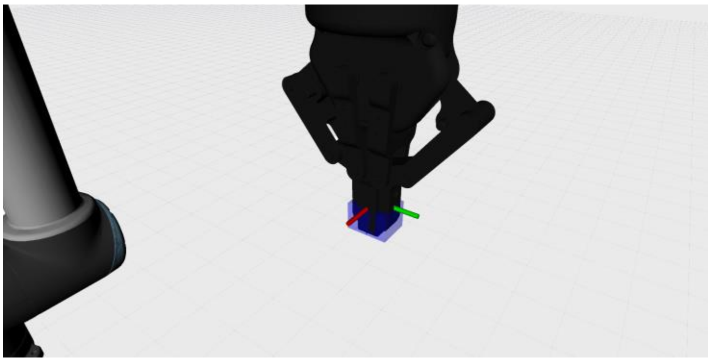
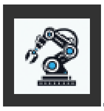


Fig 14. Collision bounding box over the fingertips.

To run a example case, just type click in its icon .

In the RQT pop-up, send the goal request by selecting with right button the Grasp Estimation Skill>Publish Selected Once (Fig 15.).

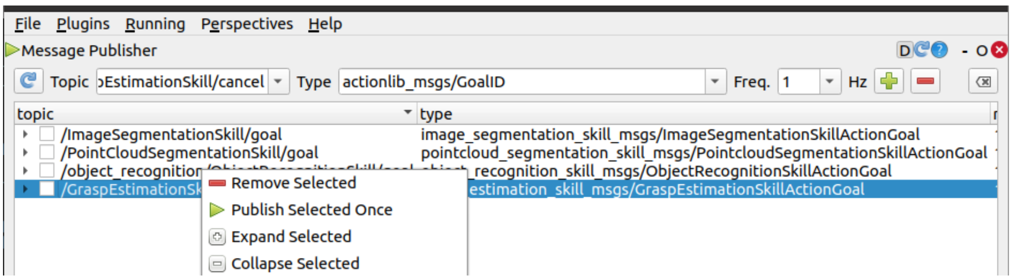


Fig 15. Sending goal request to Grasping Estimation pipeline.

Once processed, you can navigate the movement by pressing ENTER in the terminal.

Exercise 5:
Notice that the movement is not safe at all; therefore, configure the pipeline by selecting the proper sequence of heuristics in the Grasping Estimation configuration file “config_ge” located in Desktop.

TIP: We don’t want robots to rotate too much, do we?

After the end, close all the windows.

3. Build a Complete Mission

Once everything is adjusted, the Task Manager will be used to build the overall mission and synchronise the action requests. To design a complete solution, the user must build the blocks in the QT visual interface. Each block corresponds to an action. The mission will be later embedded into the robot and executed.

Exercise 6:
Open the file “mission.scxmll” , in Desktop, by right-clicking and opening with QtCreator. Therefore, complete the mission to perform the object recognition, grasping estimation and arm movement using the already inserted blocks in the template.

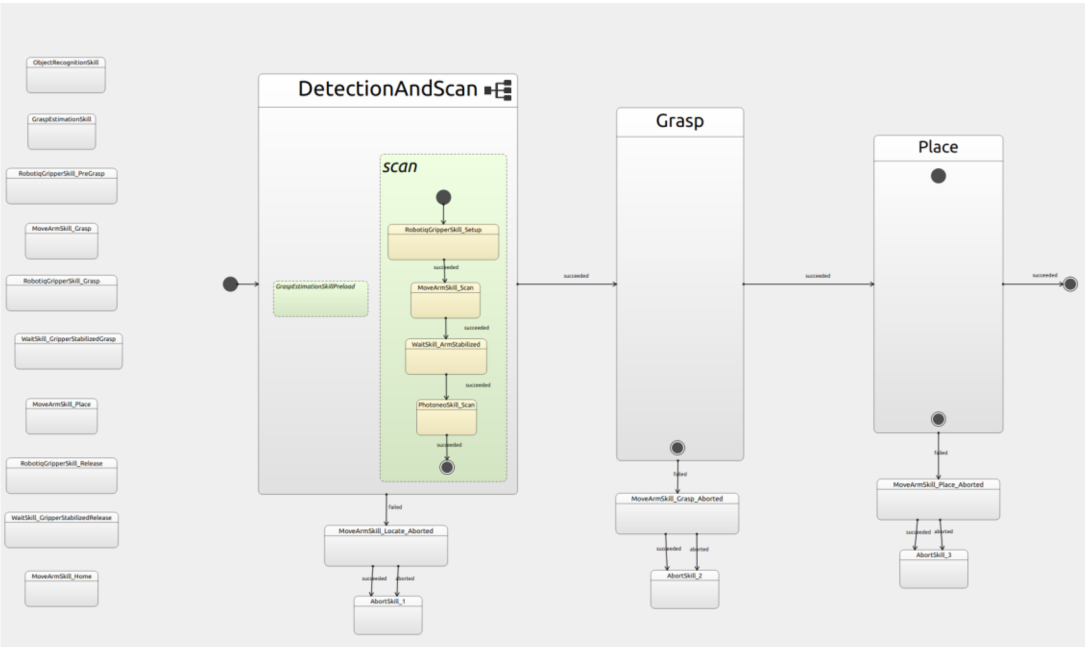


Fig 16. Building the mission.

4. Conclusion

In the current course, we introduced how to develop a custom picking solution using the system developed in the Mari4 Yard project. The course provides a brief review of the techniques, focusing on describing the necessity of each step during a bin-picking process involving robotics.

The first part of the course is dedicated to introducing the Mari4 Yard project. This system was developed to automate the task of picking, which can result in greater efficiency and accuracy.

Next, the course describes various techniques that can be used to develop a custom-picking solution. This includes discussing different algorithms that can be used and challenges that may arise during the development process.

The subsequent part of the course explains each step of the robotic bin-picking process. This includes everything from identifying the item to be picked to handling it and placing it at the desired location. Each step is discussed in detail, emphasizing its importance for the overall success of the process.

Finally, the last step of the course involves running all processing solutions on a real robot. This will take place at INESC TEC's iiLab laboratory under the guidance of an instructor.

Mobile Manipulator for Internal Logistics

Module 5: AR-based Human-Robot Interaction

Marcelo Petry, PhD.

Senior Researcher

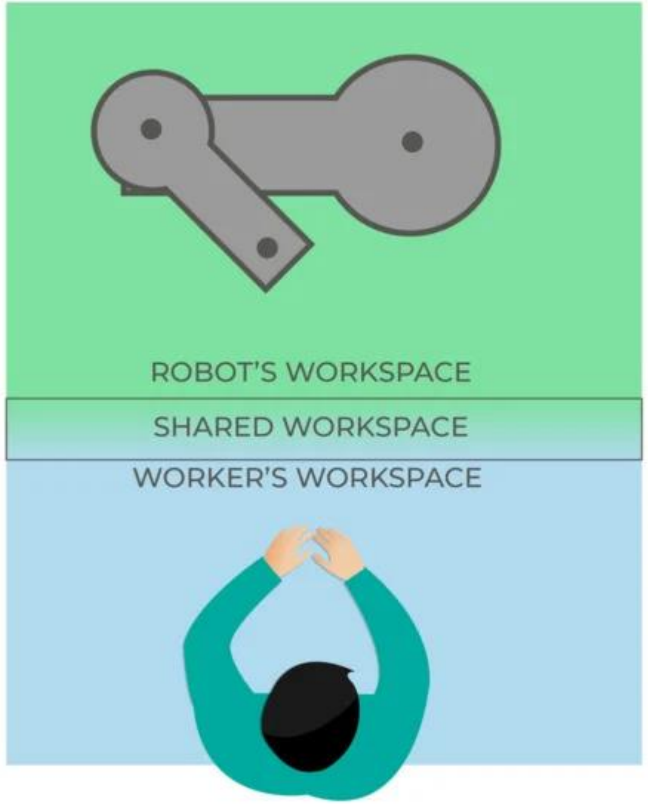
June 12, 2024

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Human-robot Interaction

Several terms can be prone to misguided and inconsistent use in industrial technical communication:

- Robot’s workspace: space that humans can’t reach
- Worker’s workspace: space that the robot can’t reach
- Shared workspace: overlapping space between the human and the robot’s reach

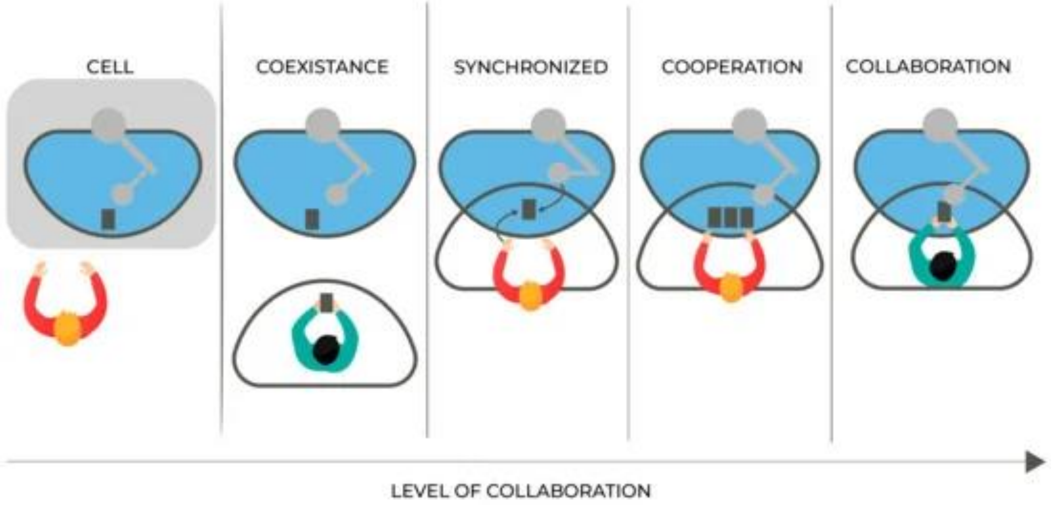


This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Human-robot Interaction

Human–Robot Interaction: general term for all forms of interaction between humans and robots

- Cell: human and robots are separated by a safety cage
- Coexistence: humans and robots are in the same environment but have separated workspaces
- Synchronized: human and robots share the workspace, but at different times
- Cooperation: human and robots share the workspace at the same time, though each element of this interaction performs a different task
- Collaboration: humans and robots share the workspace, executing the same shared task at the same time

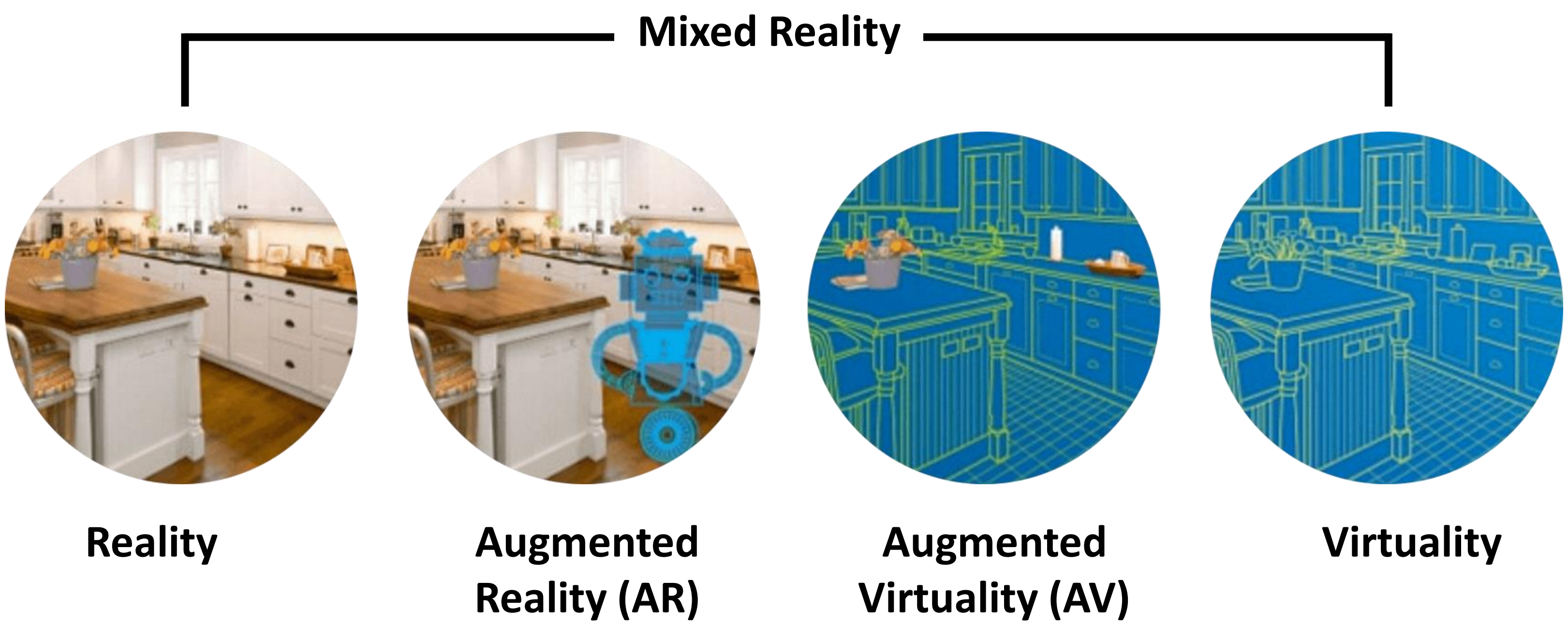


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Fundamentals of Extended Reality

This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Reality-virtuality Continuum



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Virtual Reality

Fully immerses the user in a digital environment.

- Senses:
  - Sight
  - Balance
  - Hearing
- Additional hardware:
  - Haptic gloves
  - Treadmills
  - Scent machines
- Main techs:
  - Head-mounted displays
  - Projection

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Virtual Reality

Head-mounted display paradigm:

A. Base stations

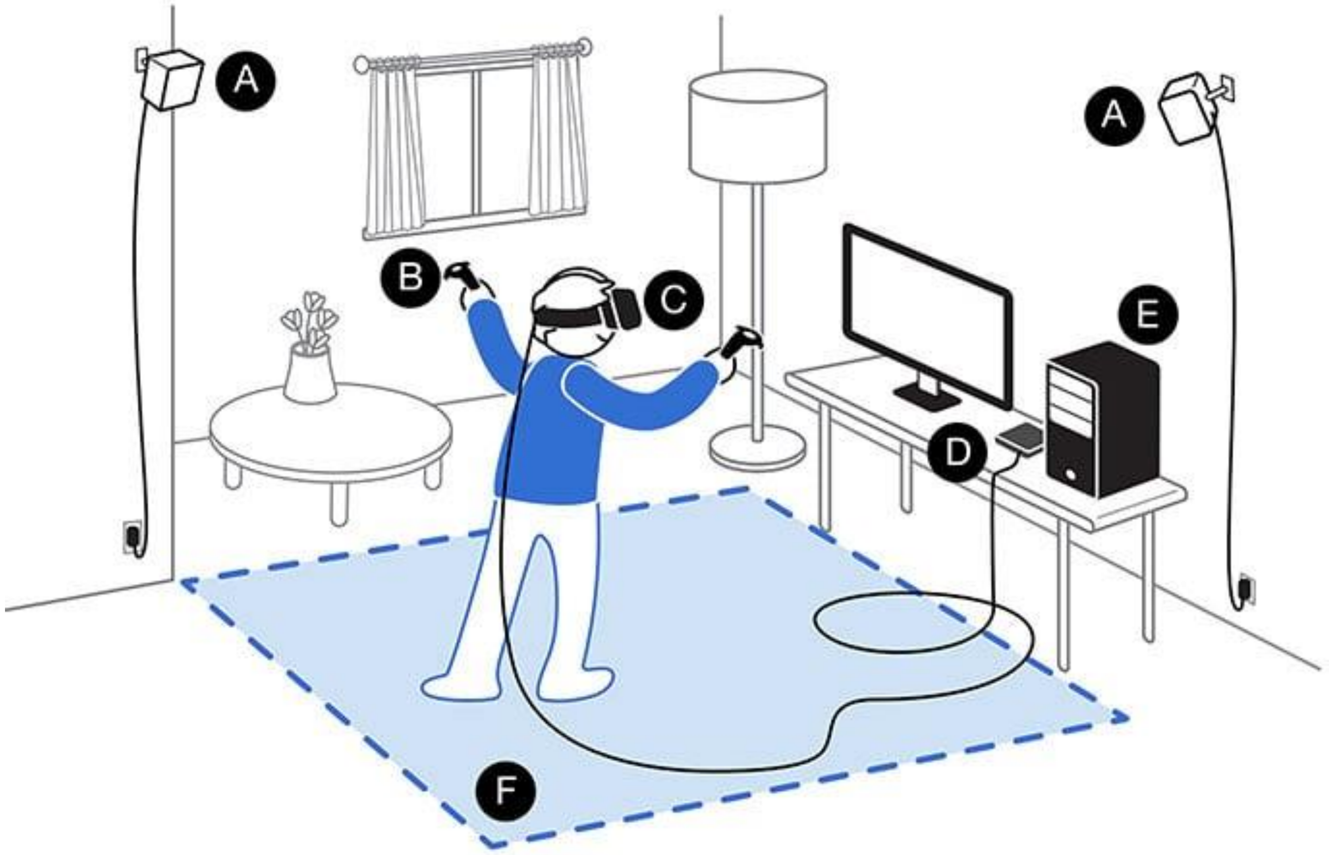
B. Hand-held controllers

C. Head-mounted display

D. Link box

E. VR computer

F. Play area

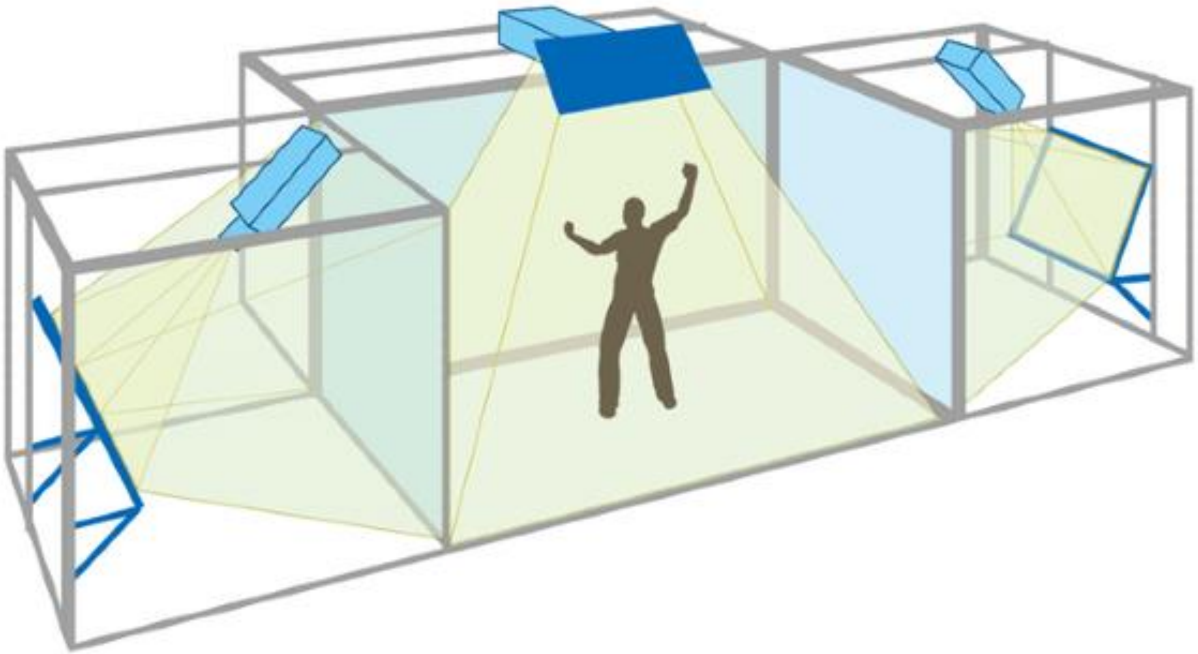


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Virtual Reality

Room-scale paradigm:

- Room-sized cube
- Projections screens
- Hand-held controllers



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Virtual Reality

Room-scale paradigm:

- Room-sized cube
- Projections screens
- Hand-held controllers
- Polarized glasses
- Tracking system

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Augmented Reality

Digital information is overlaid onto the user's view of the real world:

- Complement:
  - Sight
  - Hearing
- Main techs:
  - Head-mounted displays
  - Handheld displays
  - Projection

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Augmented Reality

Head-mount displays:

- Two displays
  - Translucid
  - Opaque

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Augmented Reality

Head-mounted displays:

- Two displays
  - Translucid
  - Opaque
- Sensors:
  - Cameras
  - Accelerometers
  - Gyroscopes
  - Depth sensors

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Augmented Reality

Hand-held displays:

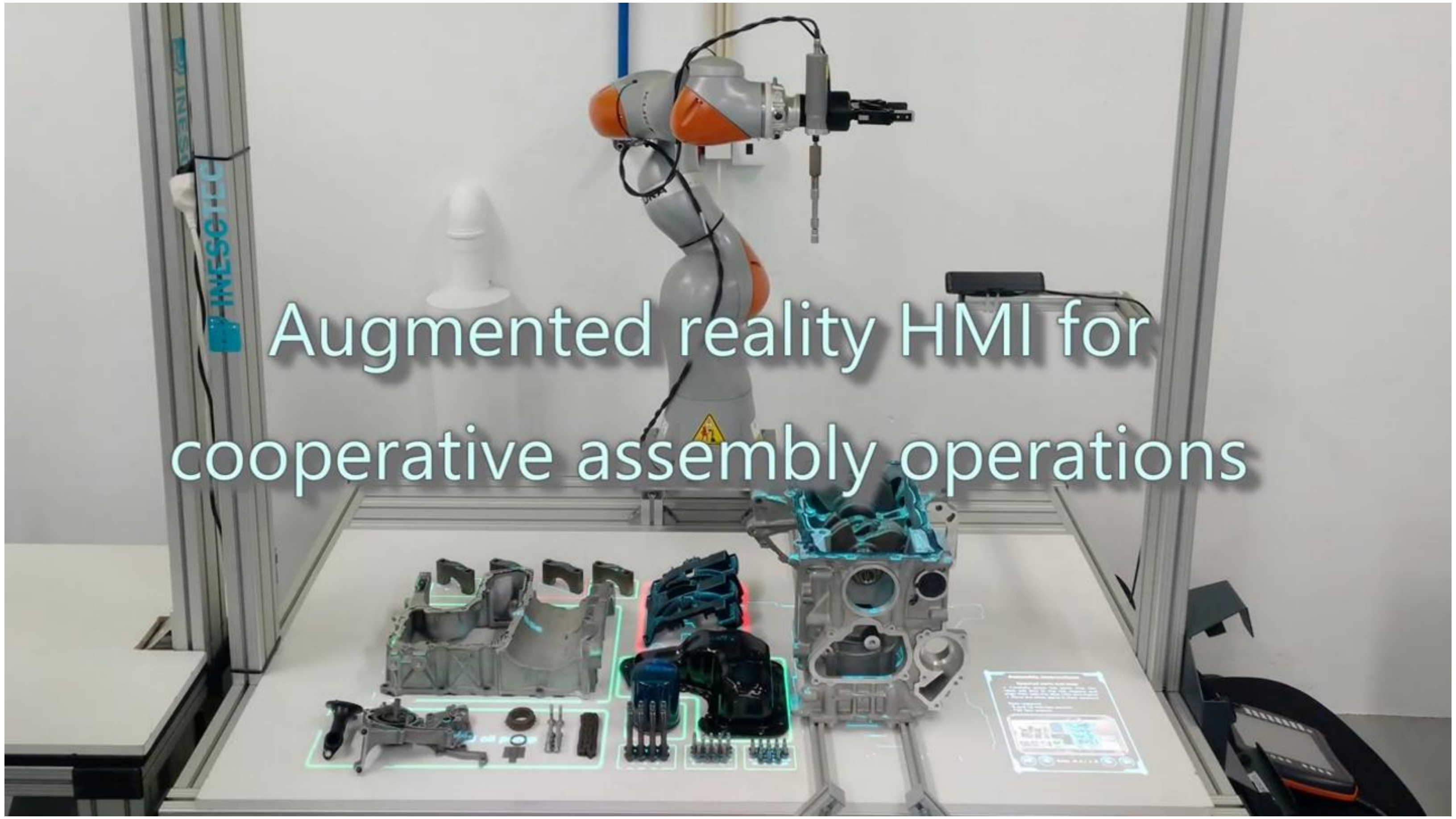
- Two displays
  - Translucid
  - Opaque
- Sensors:
  - Cameras
  - Accelerometers
  - Gyroscopes
  - Depth sensors

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Augmented Reality

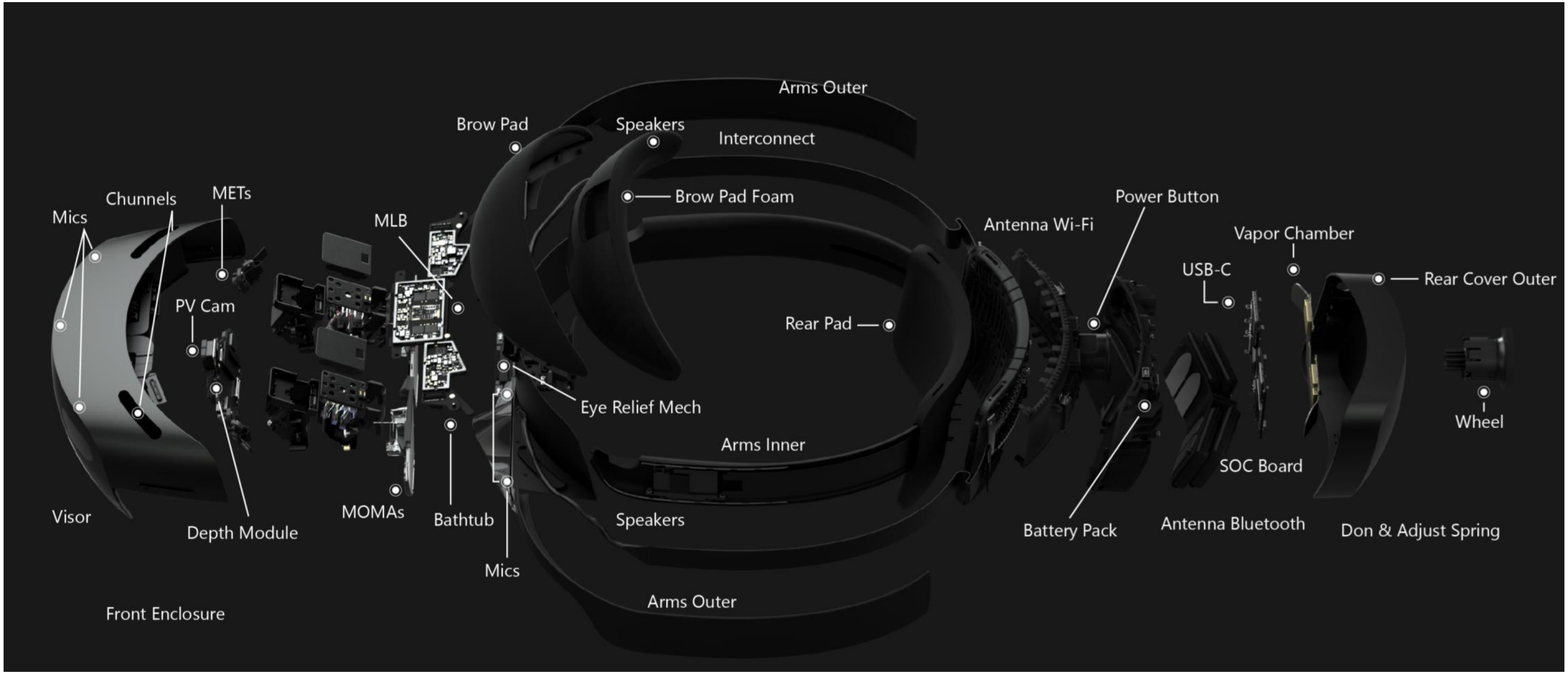
Projection:

- Video/laser projector
- Sensors:
  - Cameras
  - Depth sensors



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Microsoft Hololens 2 Overview



This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Microsoft Hololens 2 Overview

Device capabilities

Head tracking:

- 4 visible light cameras
- IMU

Eye tracking:

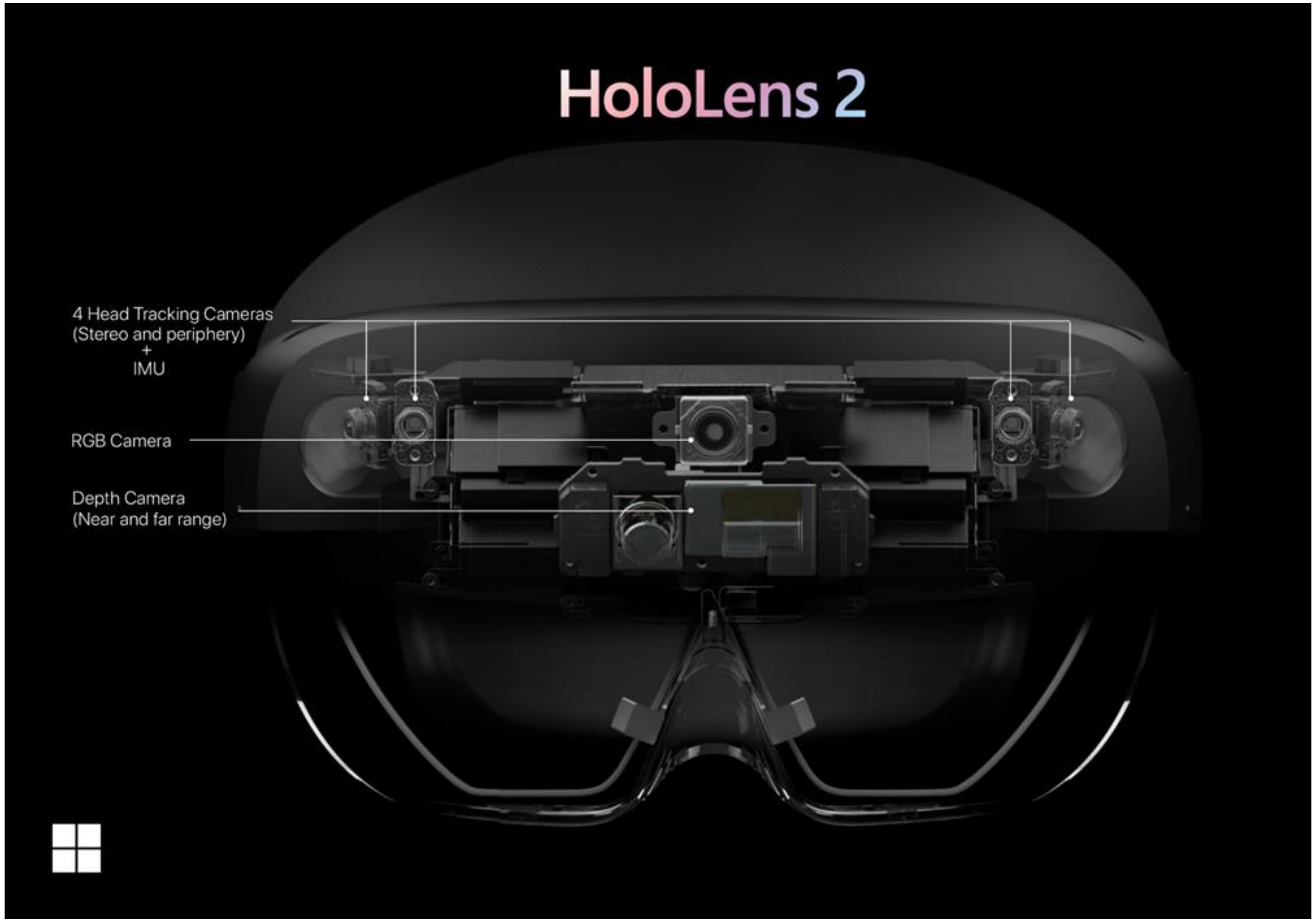
- 2 IR cameras

Hand tracking/depth:

- Time-of-Flight camera

Voice control:

- 7-microphone array



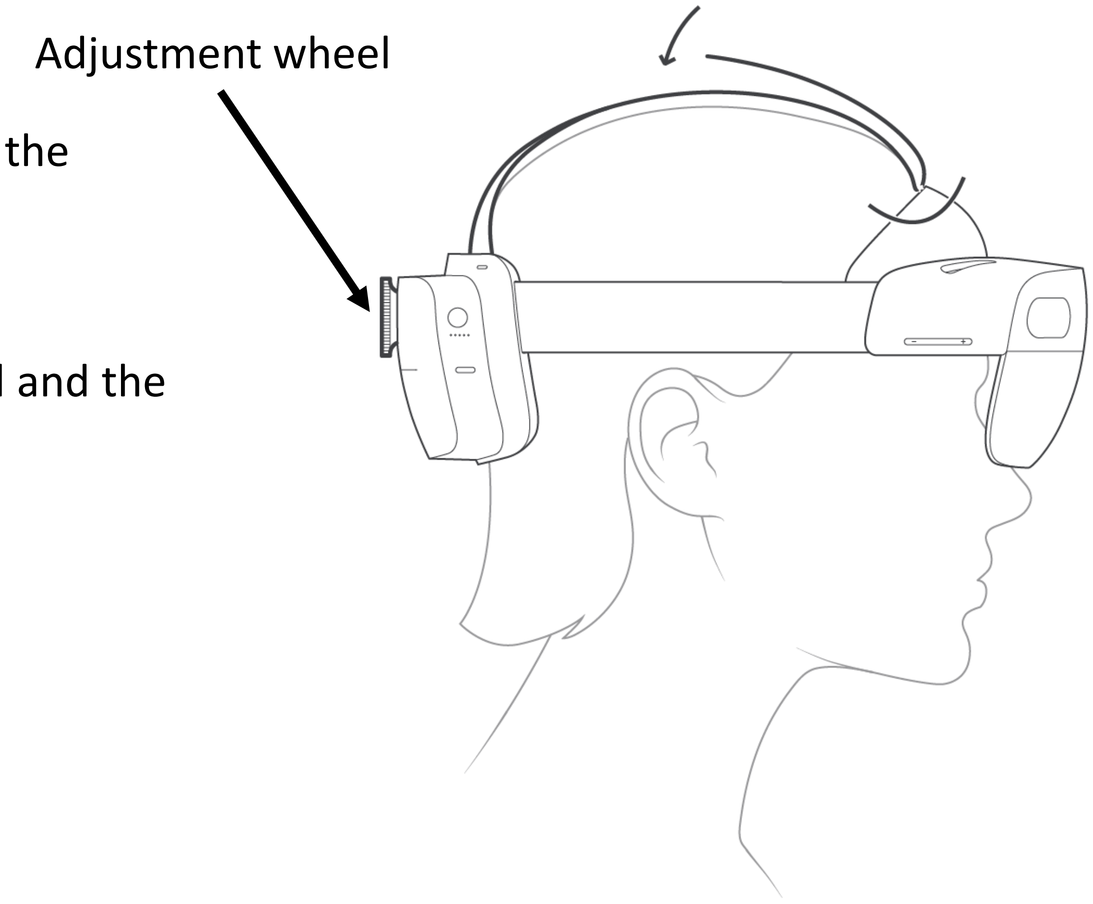
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

Adjust fit:

- Place Hololens 2 on your head
- If you wear eyeglasses, leave them on
- If necessary, extend/loose the headband by turning the adjustment wheel

*The brow pad should sit comfortably on your forehead and the back band should sit in the middle-back of your head.



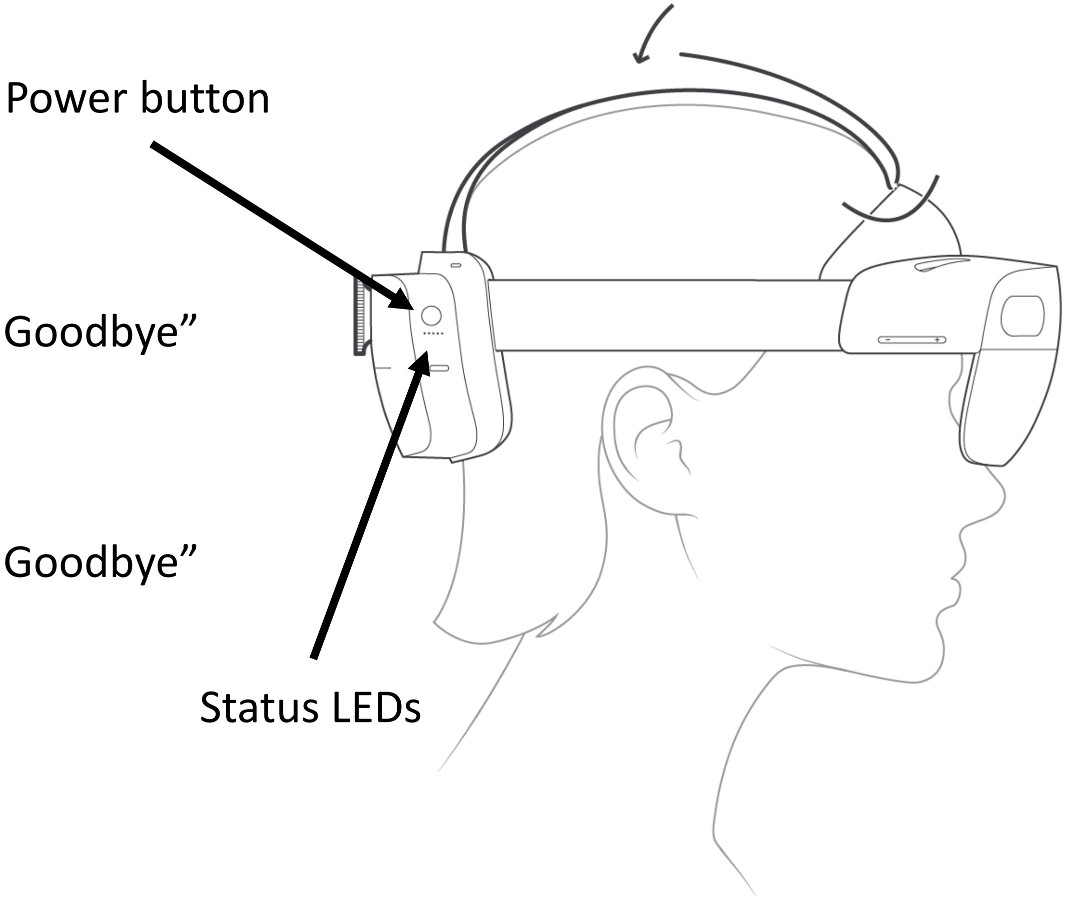
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

Turning on: single press the Power button
- All five LEDs below the Power button will turn on
- After four seconds, a sound plays

Sleep: single press the Power button
- All five LEDs turn on, then fade off one at a time
- After the lights turn off, a sound plays and the screen displays "Goodbye”

Turning off: press and hold the Power button for 5s
- All five LEDs turn on, then fade off one at a time
- After the lights turn off, a sound plays and the screen displays "Goodbye”



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

The first time you use HoloLens 2 you may be asked to calibrate HoloLens to your eyes:

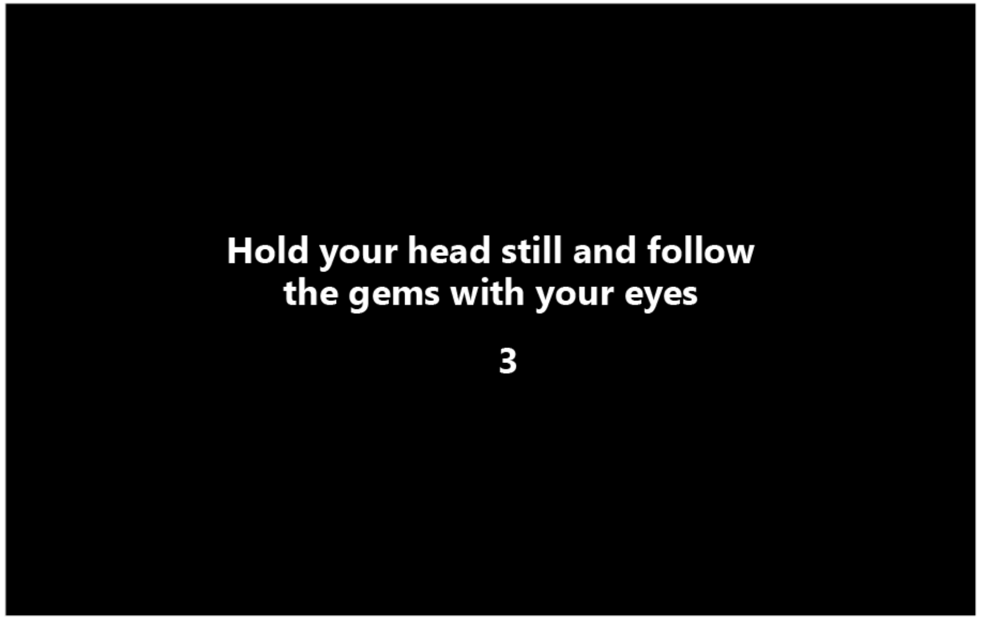
- First, adjust the visor

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

The first time you use HoloLens 2 you may be asked to calibrate HoloLens to your eyes:

- To calibrate, you'll look at a set of targets (referred to as gems)
- Try not to move your head



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

The first time you use HoloLens 2 you may be asked to calibrate HoloLens to your eyes:

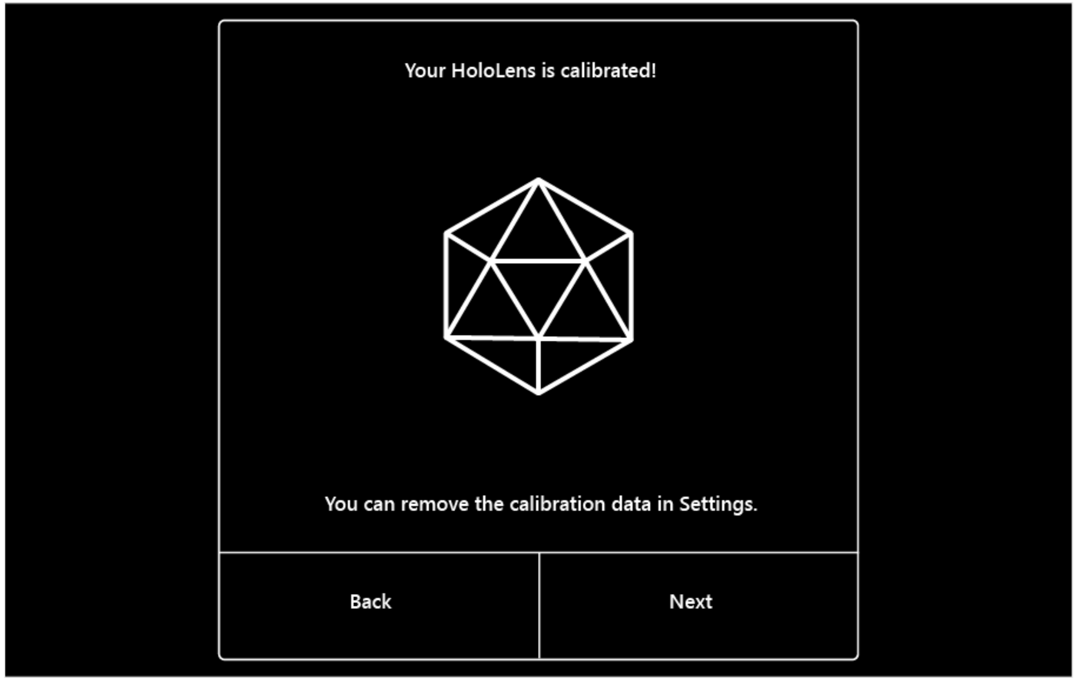
- Focus on the gems instead of other objects in the room
- HoloLens uses this process to learn about your eye position so that it can better render your holographic world

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Setting Up a New User on Hololens 2

The first time you use HoloLens 2 you may be asked to calibrate HoloLens to your eyes:

- After calibration, holograms will appear correctly even as the visor shifts on your head.
- Calibration information is stored locally on the device and is not associated with any account information.

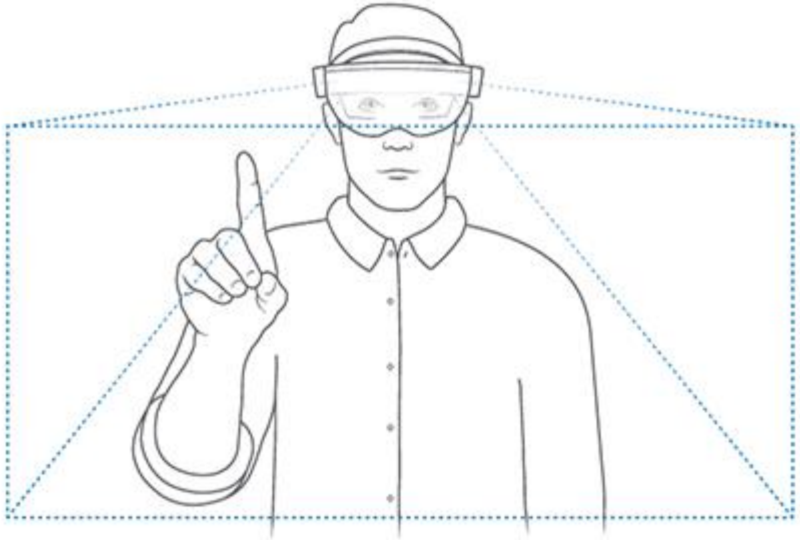


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand tracking

- Hololens use the Time-of-Flight (ToF) camera and pre-trained deep learning models to detect the user's hands and gestures
- ToF field of view (120°x120°): Keep your hands inside that frame, or HoloLens won't see them

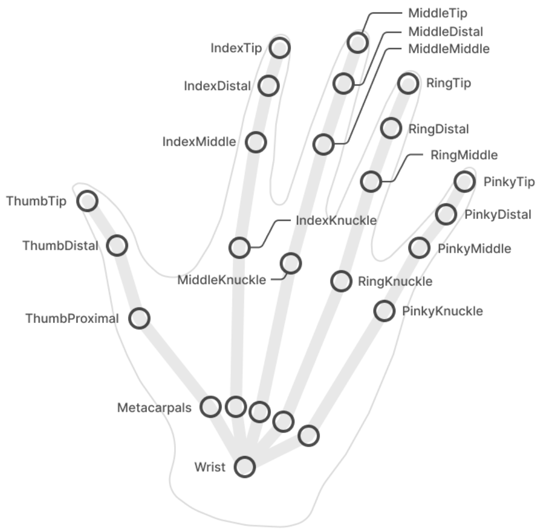


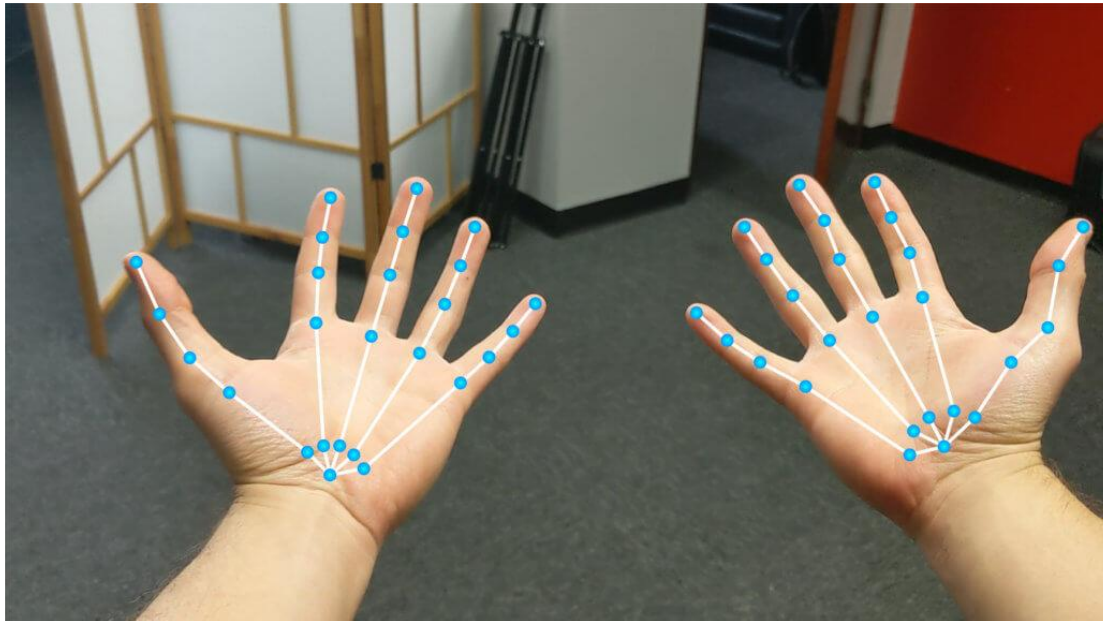
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand tracking

- In addition to providing the location of your palm, the equipment also provides estimates of 25 other points of interest.



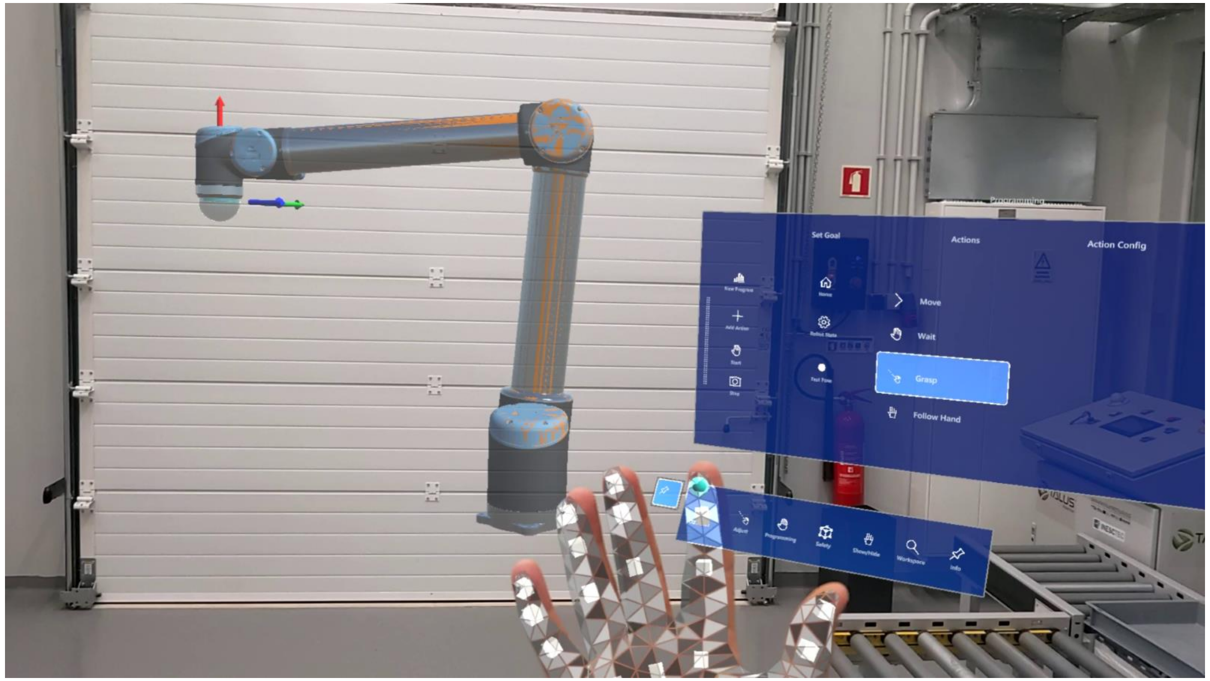


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand tracking

- When the hand is recognized Hololens may overlay a mesh over the user hand

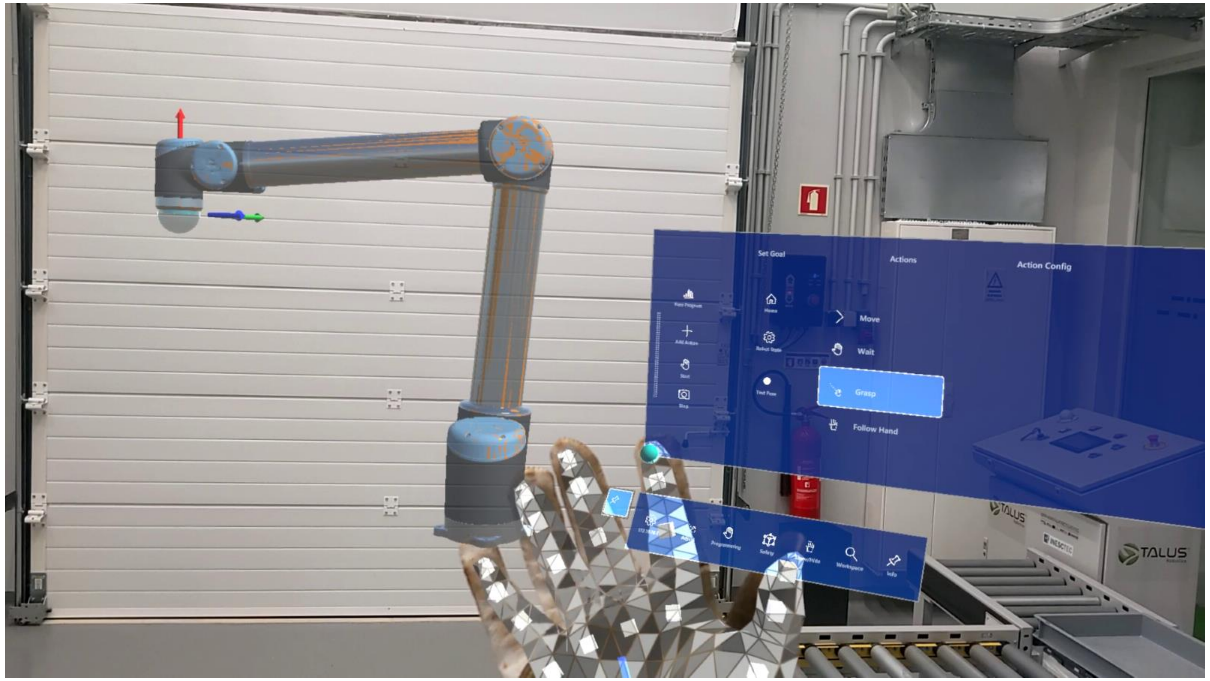


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand tracking

- The algorithm is able to detect hands even when wearing welding gloves

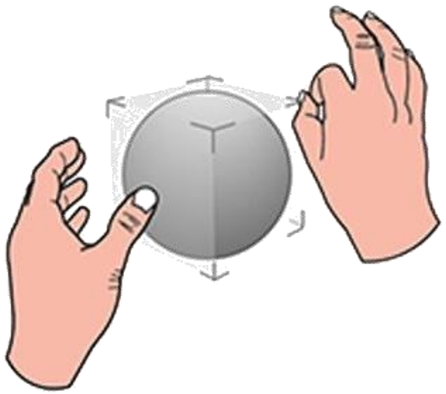


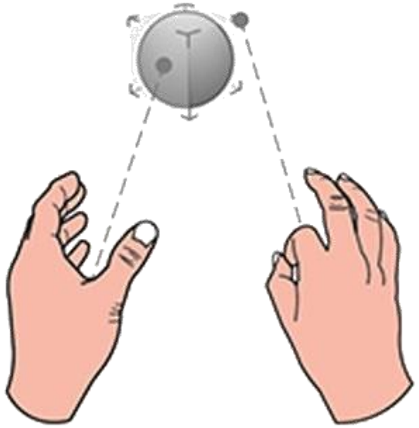
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Touching Holograms

- Direct manipulation: interaction with close holograms
- Hand rays: interaction with holograms out of reach



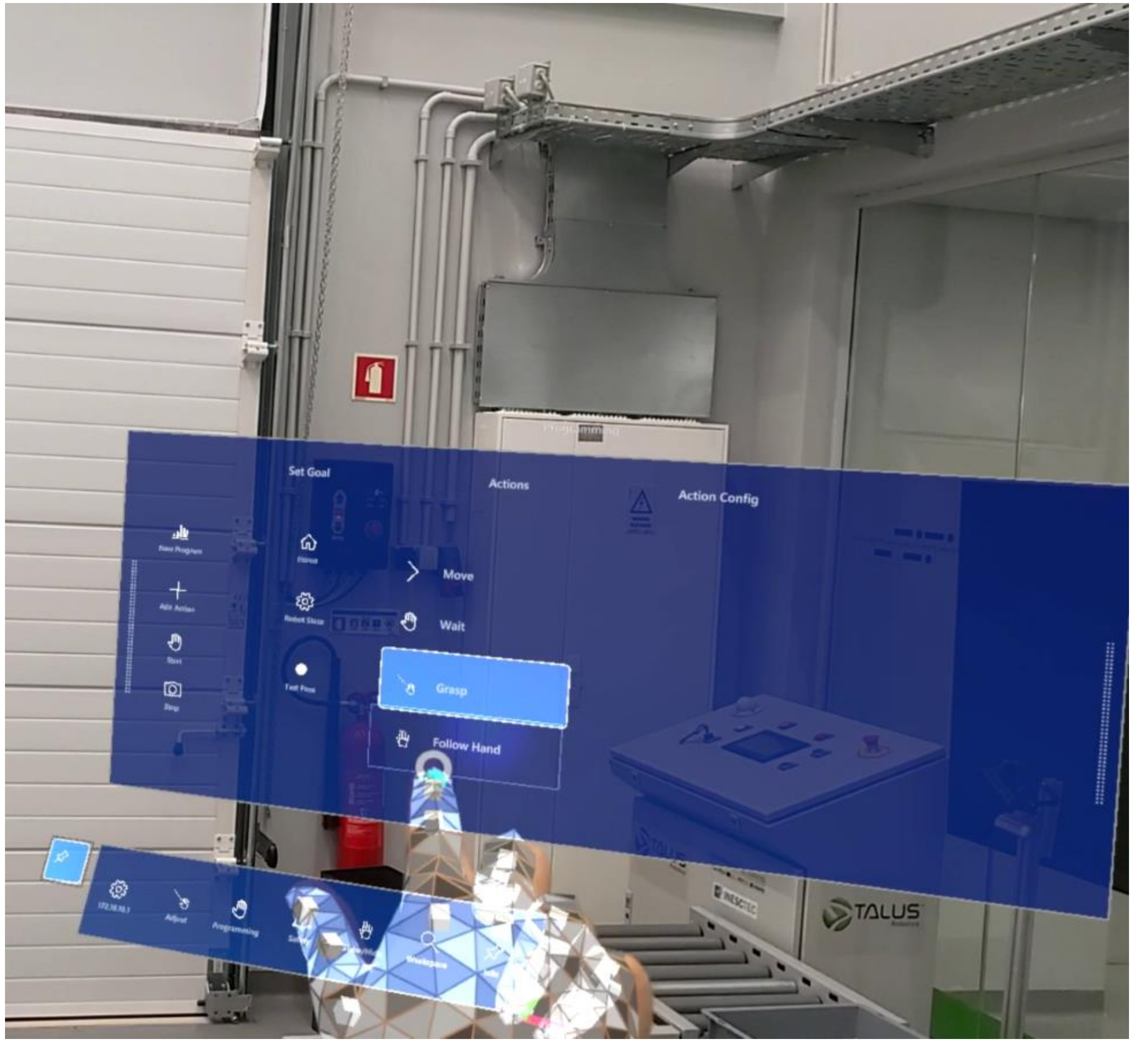


This project has received funding from
the European Union's Horizon 2020
research and innovation programme
under grant agreement No 101006798

Using Hololens 2

Direct manipulation – How to

- Bring your hand close to a hologram
- A ring (touch cursor) will be projected on the tip of your index finger

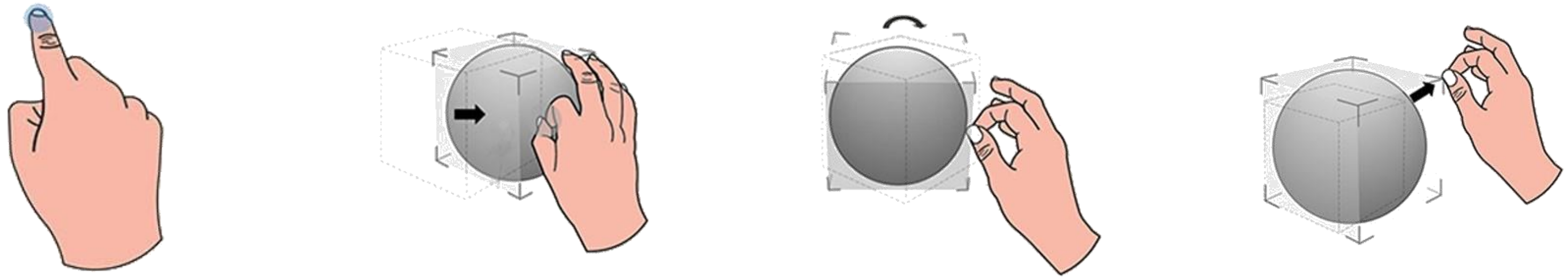


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Direct manipulation – How to

- **Select** something: simply **tap** (“touch”) it with the touch cursor
- **Scroll** content: **swipe** on the surface of the content with your finger (just like you're using a touchscreen)
- **Grab** a hologram: pinch your **thumb** and **index finger** together on the hologram and hold
- Use the **grab gesture** to move, resize, and rotate 3D objects



This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand rays – How to

- When there are no holograms near your hands, the **touch cursor** will hide automatically and **hand rays** will appear from the palm of your hands.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand rays – How to Air tap:

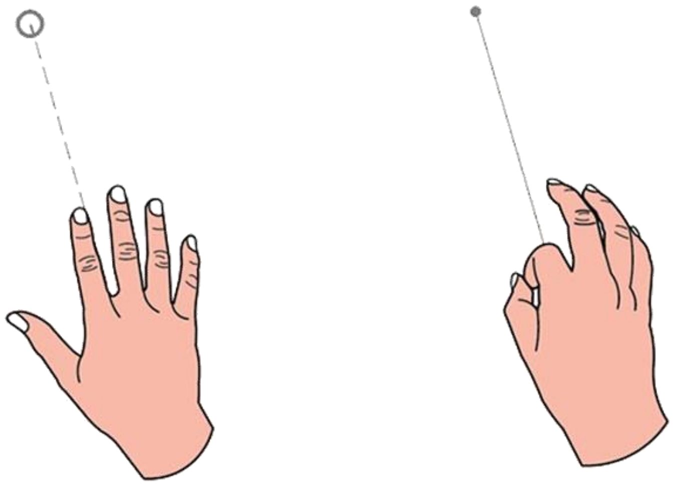
1. Use a hand ray to target the item
2. The touch cursor is displayed when the ray hits a Hologram
3. Point your index finger straight up toward the ceiling
4. Pinch your thumb and index finger together and then quickly release them

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Hand rays – How to

- **Select** a hologram: target the hologram with your hand ray and air tap
- **Grab** a hologram: target the hologram with your hand ray, then air tap and hold
- **Scroll**: air tap and hold on the content, then move your hand ray up/down or side to side



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Using Hololens 2

Start Gesture: open the start (main) menu of the HoloLens

- Two-handed:
  1. Hold your hand with your palm facing you
  2. The **Start icon** will appear over your inner wrist
  3. Tap this icon using your other hand

- One-handed:
  1. Hold your hand with your palm facing you
  2. Look at the Start icon on your inner wrist
  3. Pinch your thumb and index finger together

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Hololens 2 app for HRI

Features:

- Hands and head tracking
- Natural gestures

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

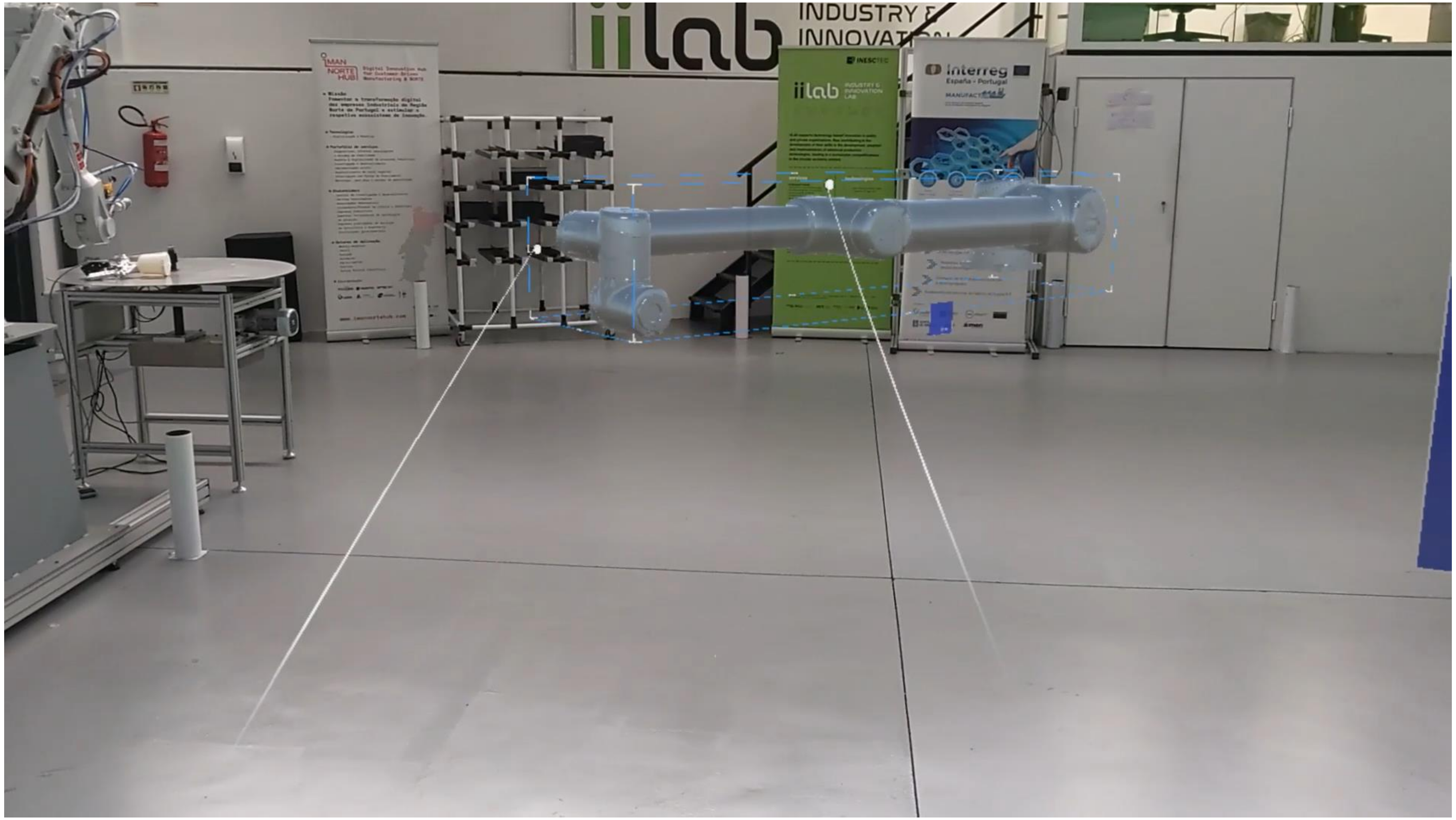
- Connection with real/simulated robots
- Multicell operation

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Hologram manipulation

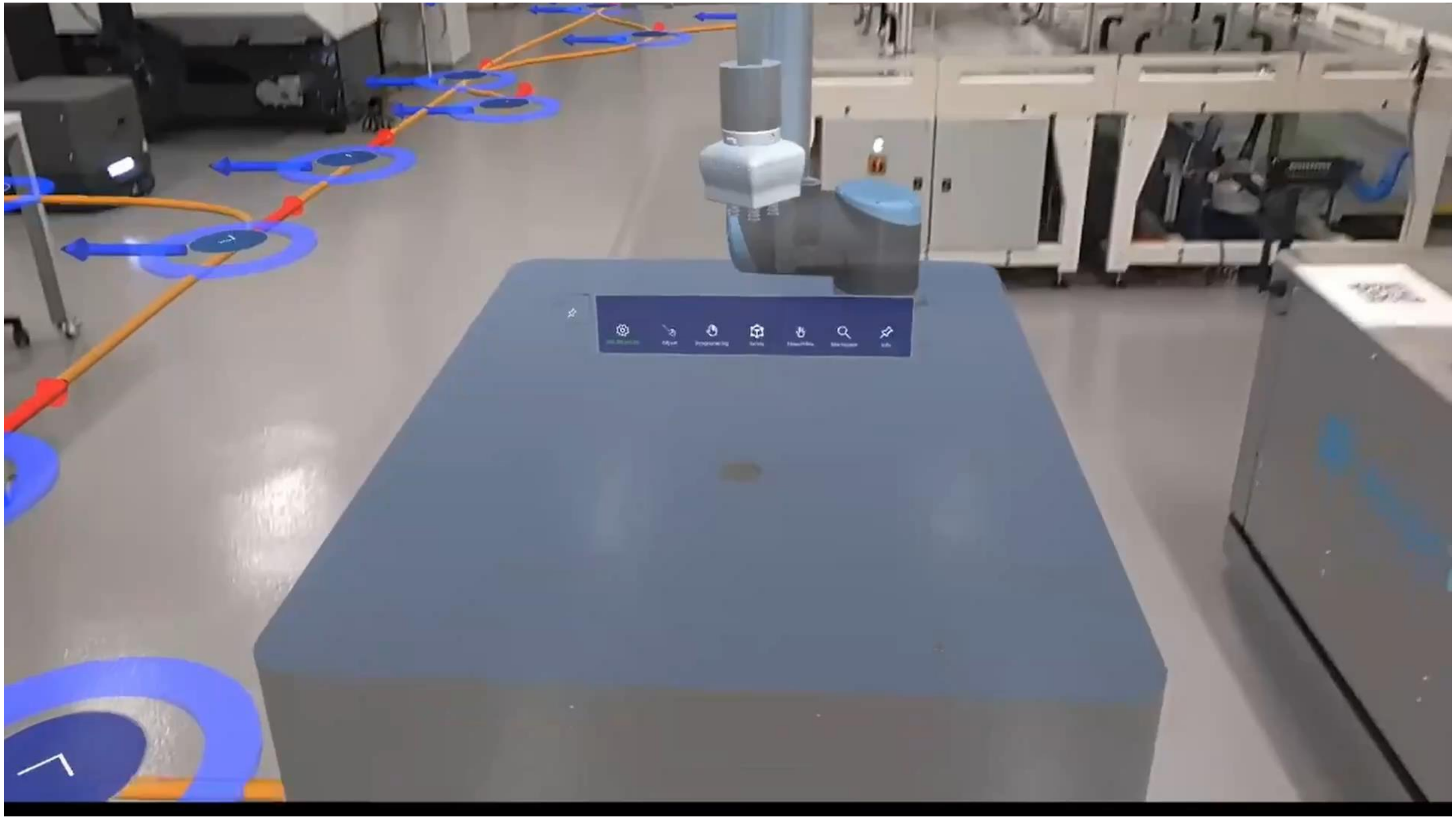


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Align the real and digital worlds automatically



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Visualization of the robot workspace

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

MARI4YARD
MARI4ALLIANCE

XRCollab

Features:

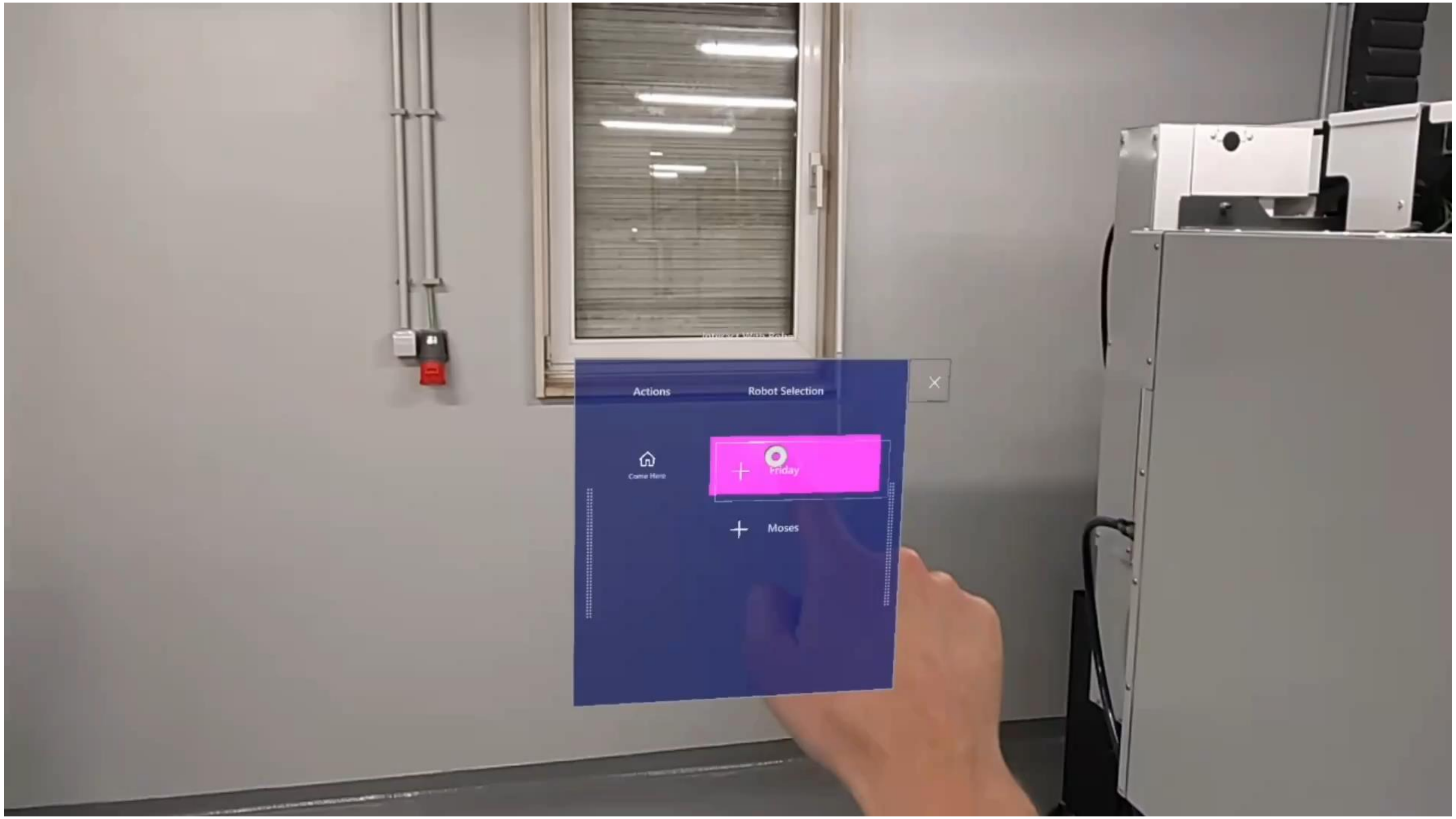
- Hand-guided programming
- Interface with robot Skills

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Visualize robot paths on the floor
- Interaction with navigation stack – Call robot to desired vertex

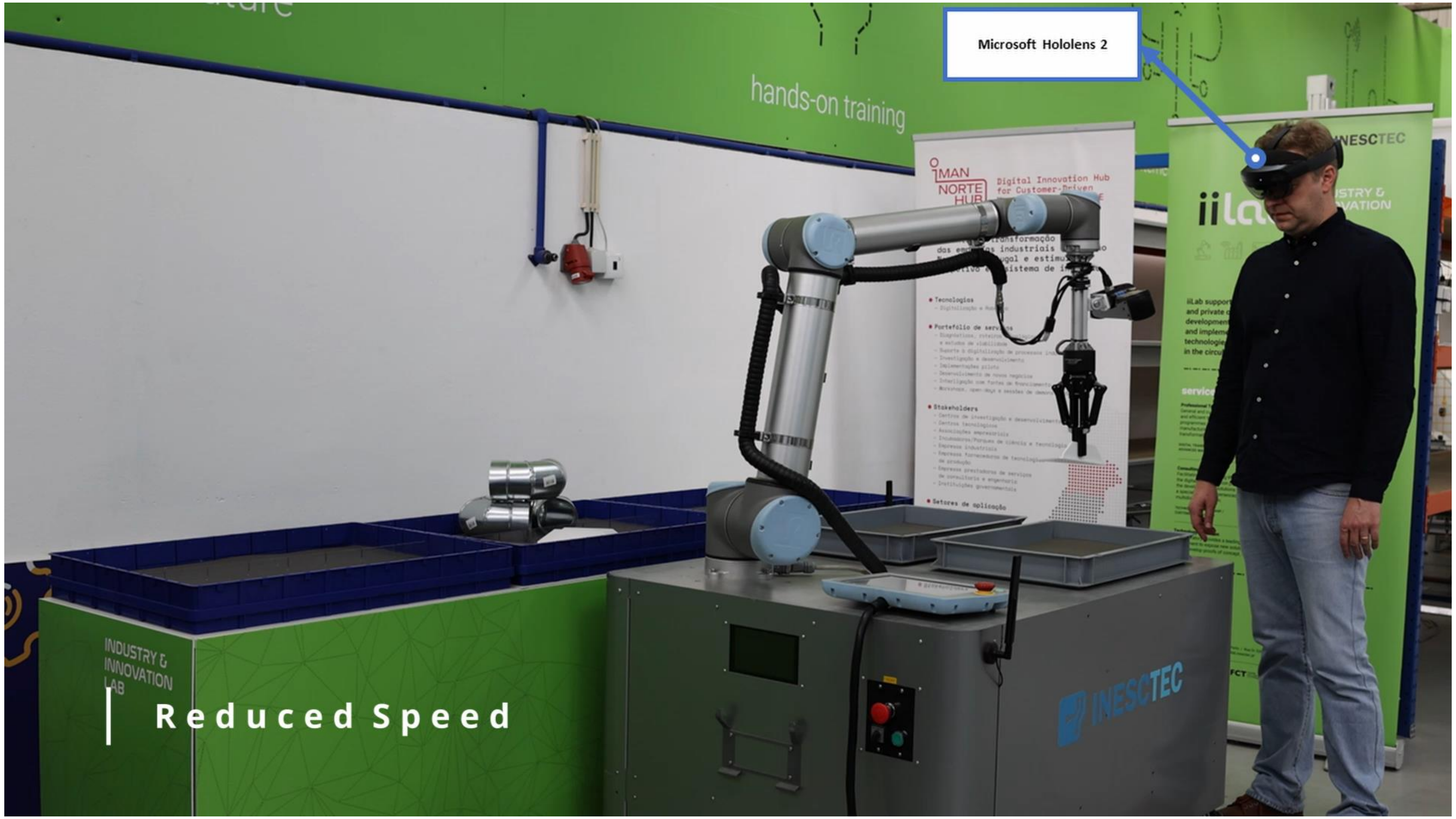


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Soft safety (not certifiable)
- Personal protection

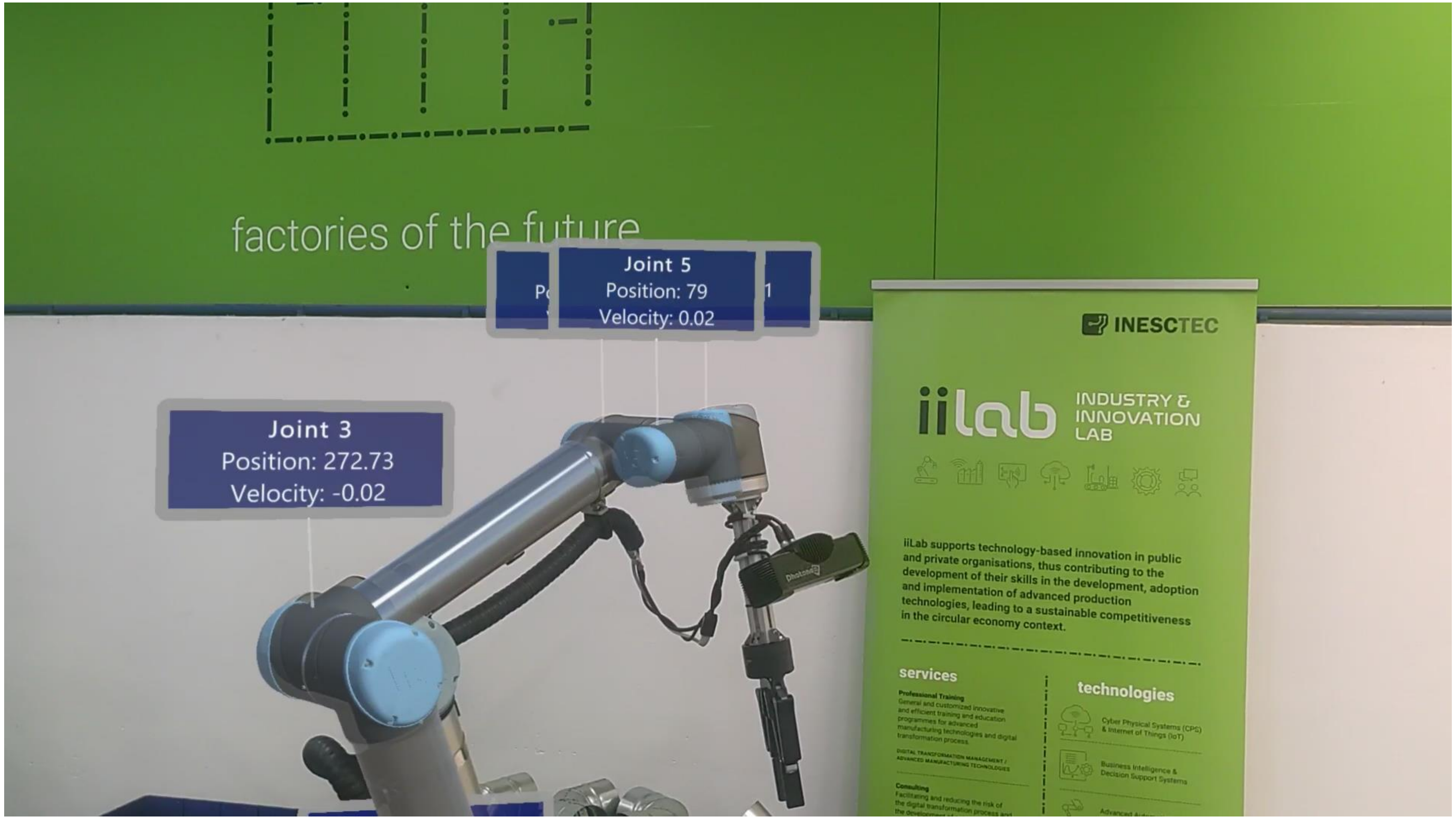


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

XRCollab

Features:

- Visualization of multidimensional data



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101006798

Thank you for your attention!

Let’s try on the robot!

MARI4YARD
MARI4ALLIANCE

Marcelo Petry, PhD. | Senior Researcher

marcelo.petry@inesctec.pt INESCTEC

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101006798