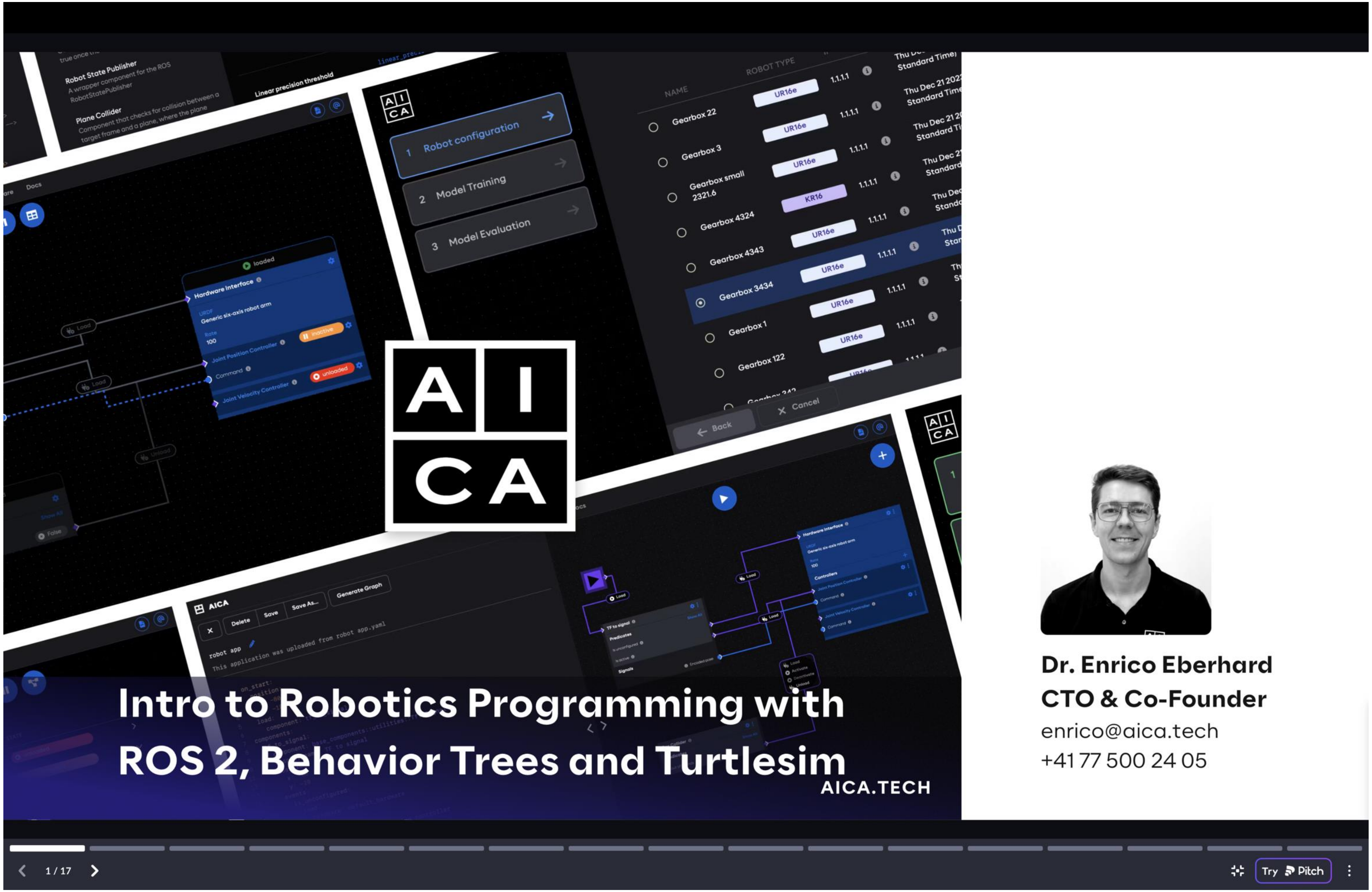
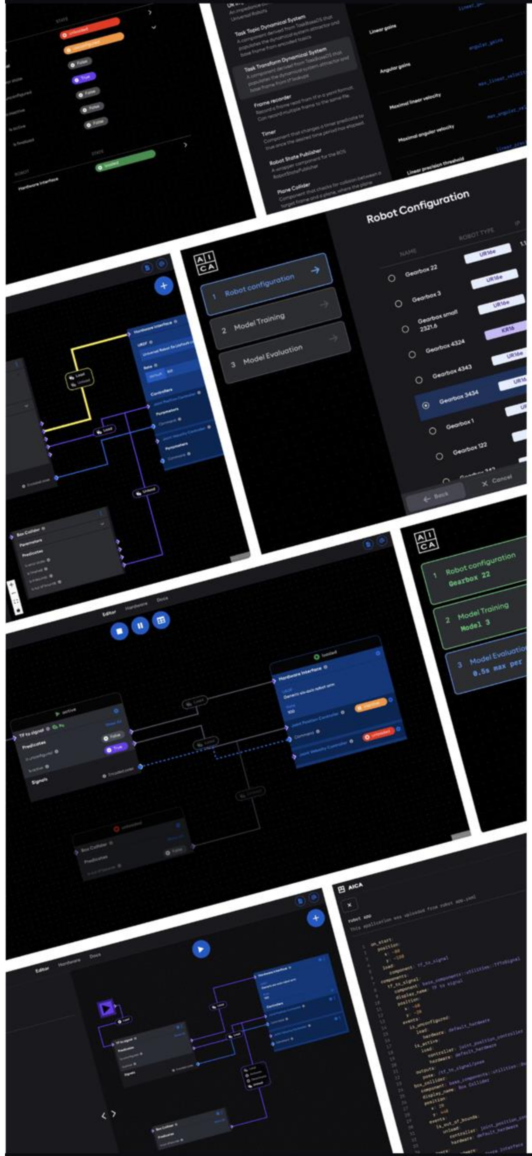


Intro to Robotics Programming with ROS 2, Behavior Trees and Turtlesim

AICA.TECH

Dr. Enrico Eberhard
CTO & Co-Founder
enrico@aica.tech
+41 77 500 24 05

Presentation and Task Content

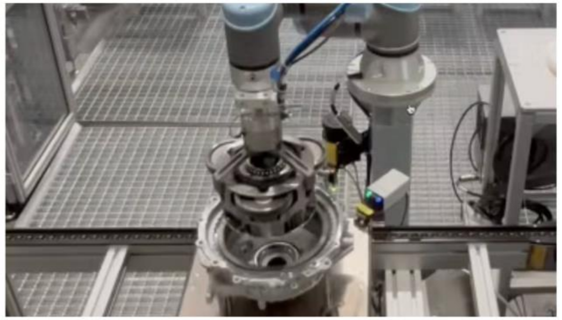


- Company introduction
- Getting into robotics programming
  - ROS 2
  - Behavior Trees
  - Turtlesim
- Task definition
  - Template repository
  - Requirements

Swiss Ecosystem of Innovation & Collaboration Built for Industrial Scale

LASA spin-off from EPFL
Delivering advanced robot control and machine learning algorithms to industry

Software first mindset
AICA's mission is to bring the best practices and toolings from the software world to industrial robotics

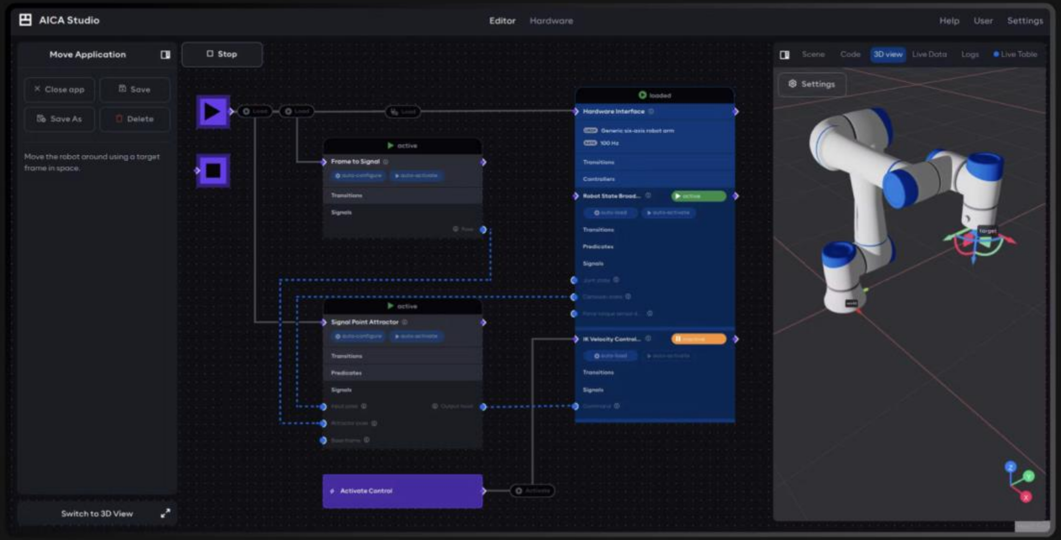


Deployment focused
AICA's software is industrial-grade and made to accelerate the development of applications, aiming for production readiness

SOLUTION

Modular software for intelligent & adaptive robot arm programming in real-time

Machine Learning. Force control. Dynamic Motion.

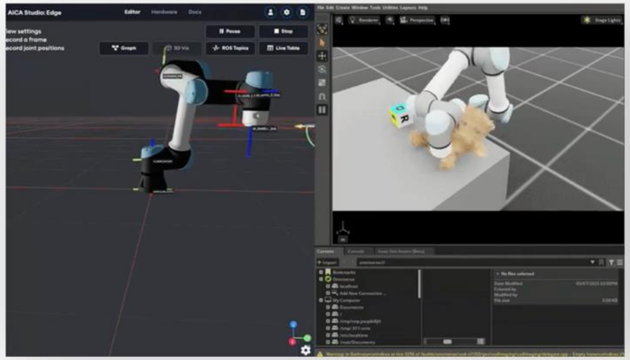


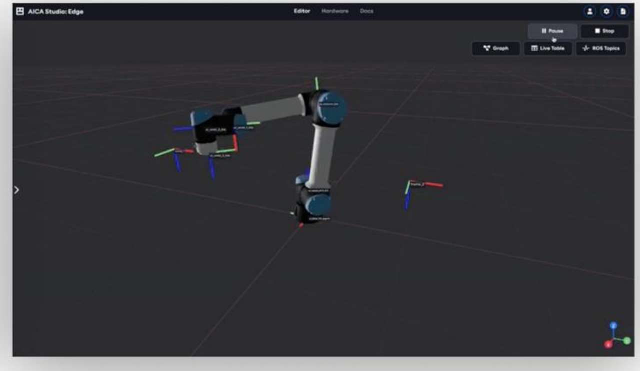
Move the robot around using a target frame in space.

WWW.AICA.TECH

How AICA Studio answers robotics bottlenecks

BENEFITS





Automate the complex & variable tasks
Force-controlled, AI & RL algorithms let robots **sense contact and adapt in real time** to flexible parts and various existing variations along with a 3D visualisation and live monitoring.

Outcome: learn behaviours instead of programming them, self-adaptation to variances.

From scripting to data flow
Changeovers become parameter edits in a low-code UI; create action sequences, define conditions, and write logic flow that is not feasible with traditional scripting.

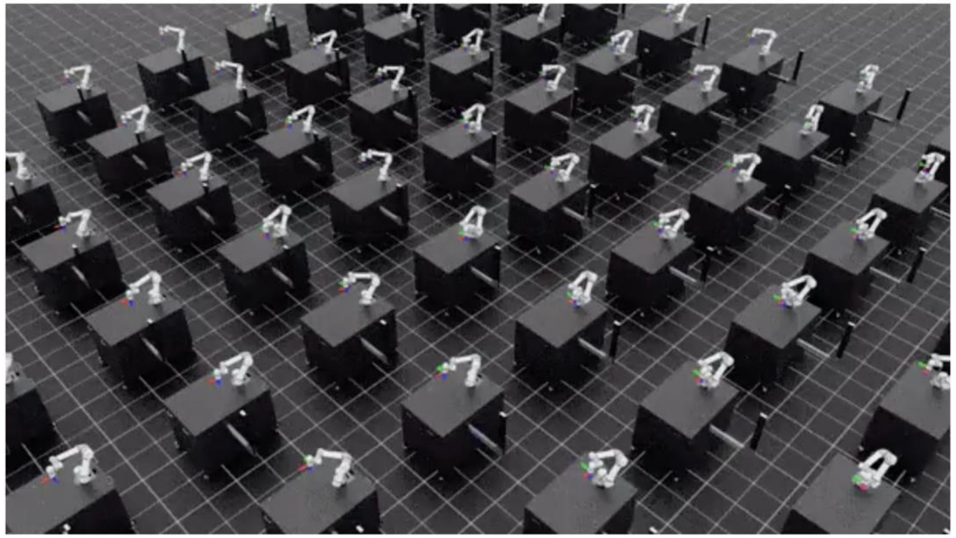
Outcome: faster adaption, new programming paradigm.

Synchronise robots, sensors, and hardware
Real-time data exchange enables the seamless integration of multiple robots, sensors and hardware to derive complex and flexible behaviours.

Outcome: develop complex applications without integration hassle.

INNOVATION

The AI in AICA: AICA stays at the cutting edge, deploying innovation to industry

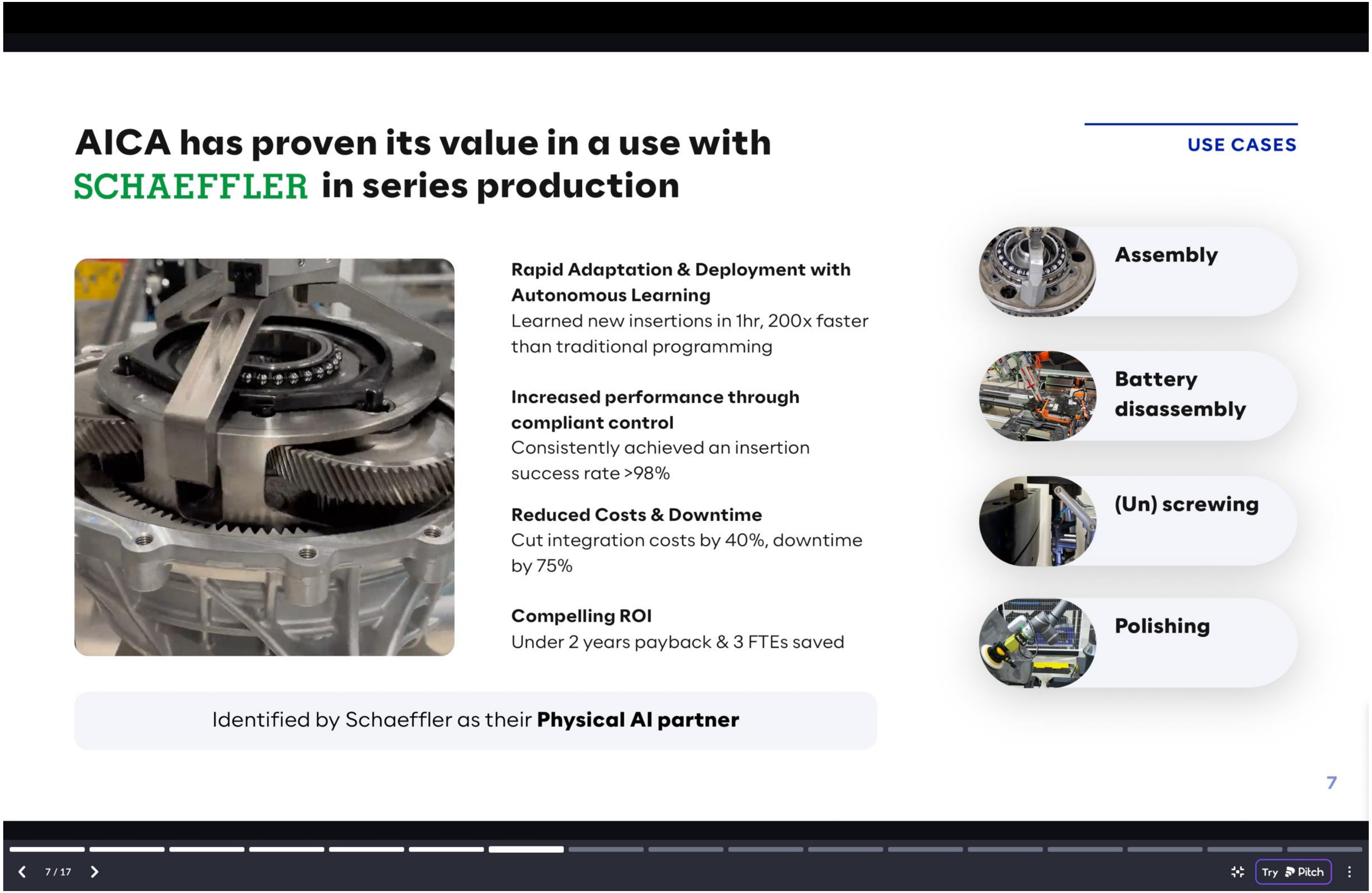


Training new models (NVIDIA Omniverse)

Deep reinforcement learning for learning bespoke vision or motion tasks in simulation, again transferred to reality within the AICA System

Deploying large models (e.g., Foundation Pose)

Large AI models trained by big companies are becoming generally available, but very few have the know-how to actually deploy them!



AICA has proven its value in a use with SCHAEFFLER in series production

USE CASES

Rapid Adaptation & Deployment with Autonomous Learning
Learned new insertions in 1hr, 200x faster than traditional programming

Increased performance through compliant control
Consistently achieved an insertion success rate >98%

Reduced Costs & Downtime
Cut integration costs by 40%, downtime by 75%

Compelling ROI
Under 2 years payback & 3 FTEs saved

Identified by Schaeffler as their **Physical AI partner**

Assembly

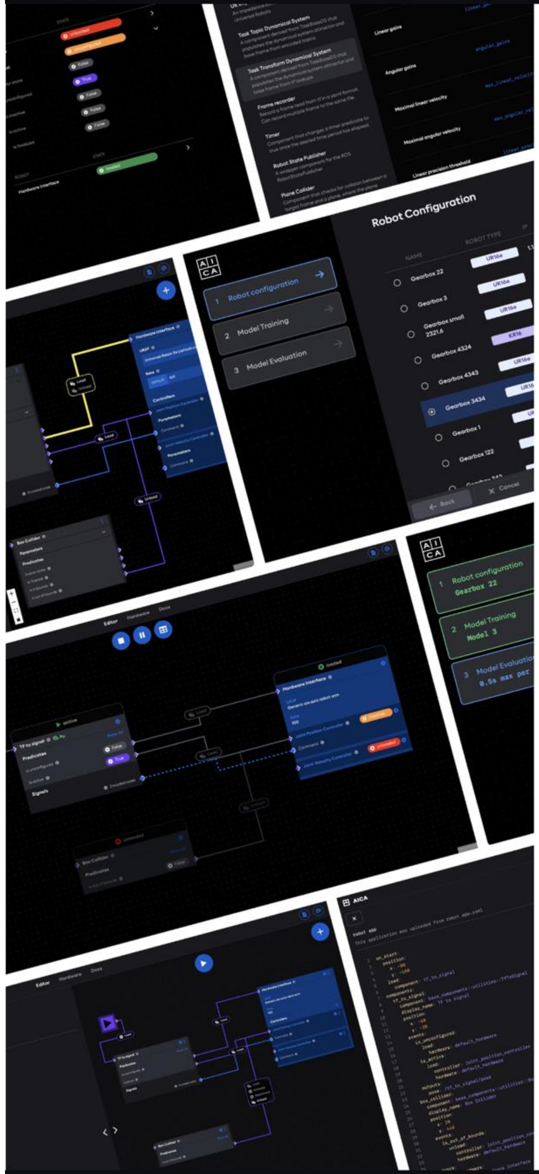
Battery disassembly

(Un) screwing

Polishing

Modular software to simplify robot arm programming for complex tasks

Machine Learning. Force control. Dynamic Motion.

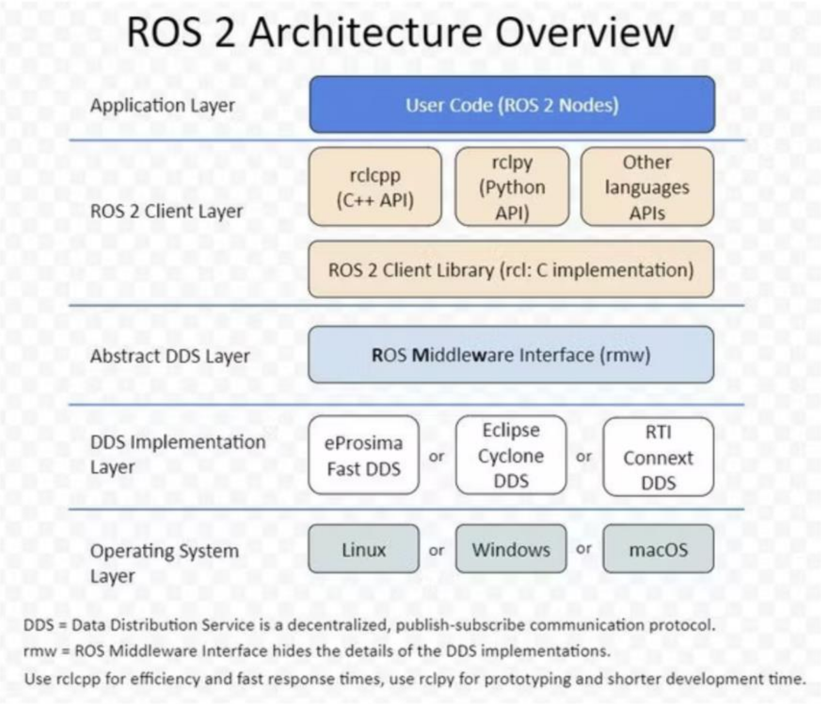


Robotics programming concepts

ROS 2

- A micro-service architecture for robotics
- Manage the execution and communication between processes
- Use existing open-source libraries for perception, planning and control

- ROS is a framework for building re-usable software components, but it does not give any opinion about ***how*** to execute a task
- We also need an orchestration framework…

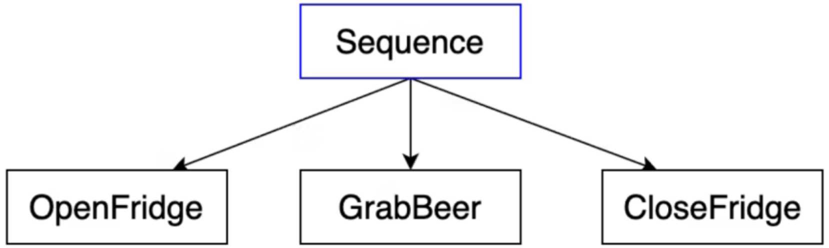


DDS = Data Distribution Service is a decentralized, publish-subscribe communication protocol.
rmw = ROS Middleware Interface hides the details of the DDS implementations.
Use rclcpp for efficiency and fast response times, use rclpy for prototyping and shorter development time.

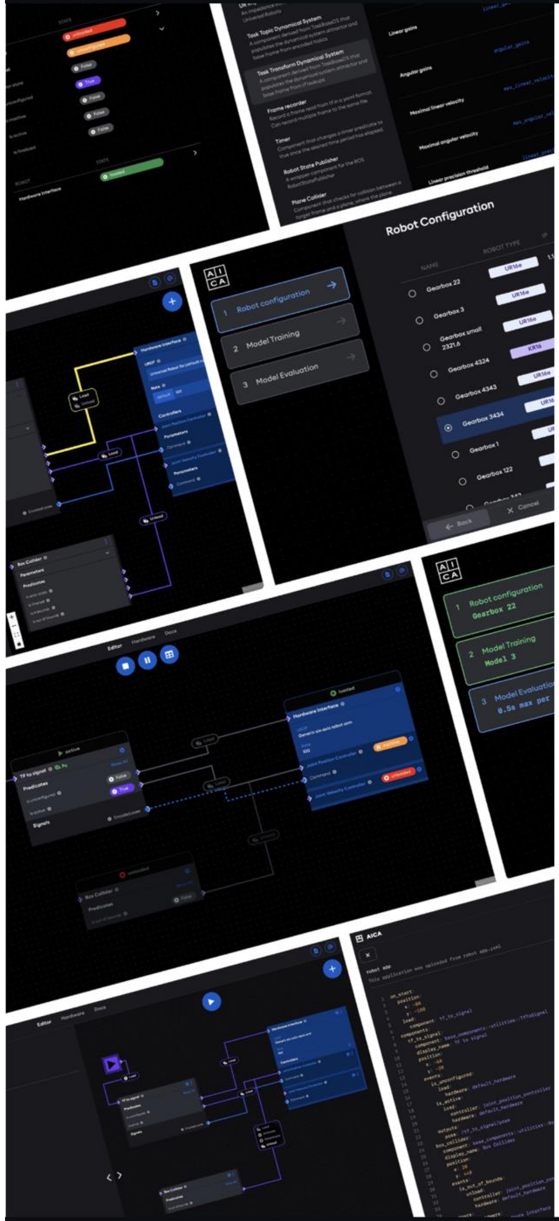
https://docs.ros.org/en/jazzy/index.html

Behavior Trees

- Structure sequences of composable actions
- Used in video games and AI before becoming popular in robotics
- More flexible than a state machine
- Allows defining tasks declaratively (e.g. in XML) to modify behavior at deploy time, not compile time



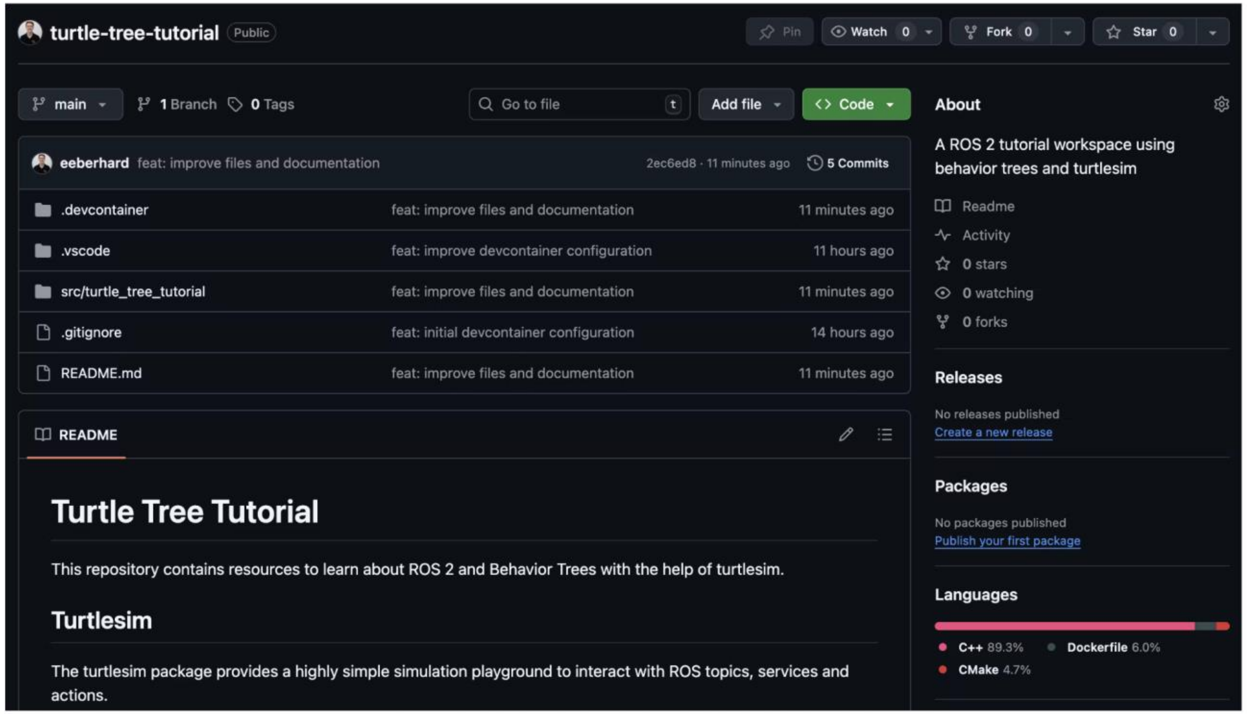
https://www.behaviortree.dev/docs/learn-the-basics/BT_basics



Task definition

Template repository

Set up a development environment using VSCode and Docker
https://github.com/eeberhard/turtle-tree-tutorial



eeberhard feat: improve files and documentation 2ec6ed8 · 11 minutes ago 5 Commits

.devcontainer	feat: improve files and documentation	11 minutes ago
.vscode	feat: improve devcontainer configuration	11 hours ago
src/turtle_tree_tutorial	feat: improve files and documentation	11 minutes ago
.gitignore	feat: initial devcontainer configuration	14 hours ago
README.md	feat: improve files and documentation	11 minutes ago

Turtle Tree Tutorial

This repository contains resources to learn about ROS 2 and Behavior Trees with the help of turtlesim.

Turtlesim

The turtlesim package provides a highly simple simulation playground to interact with ROS topics, services and actions.

About

A ROS 2 tutorial workspace using behavior trees and turtlesim

Readme
Activity
0 stars
0 watching
0 forks

Releases

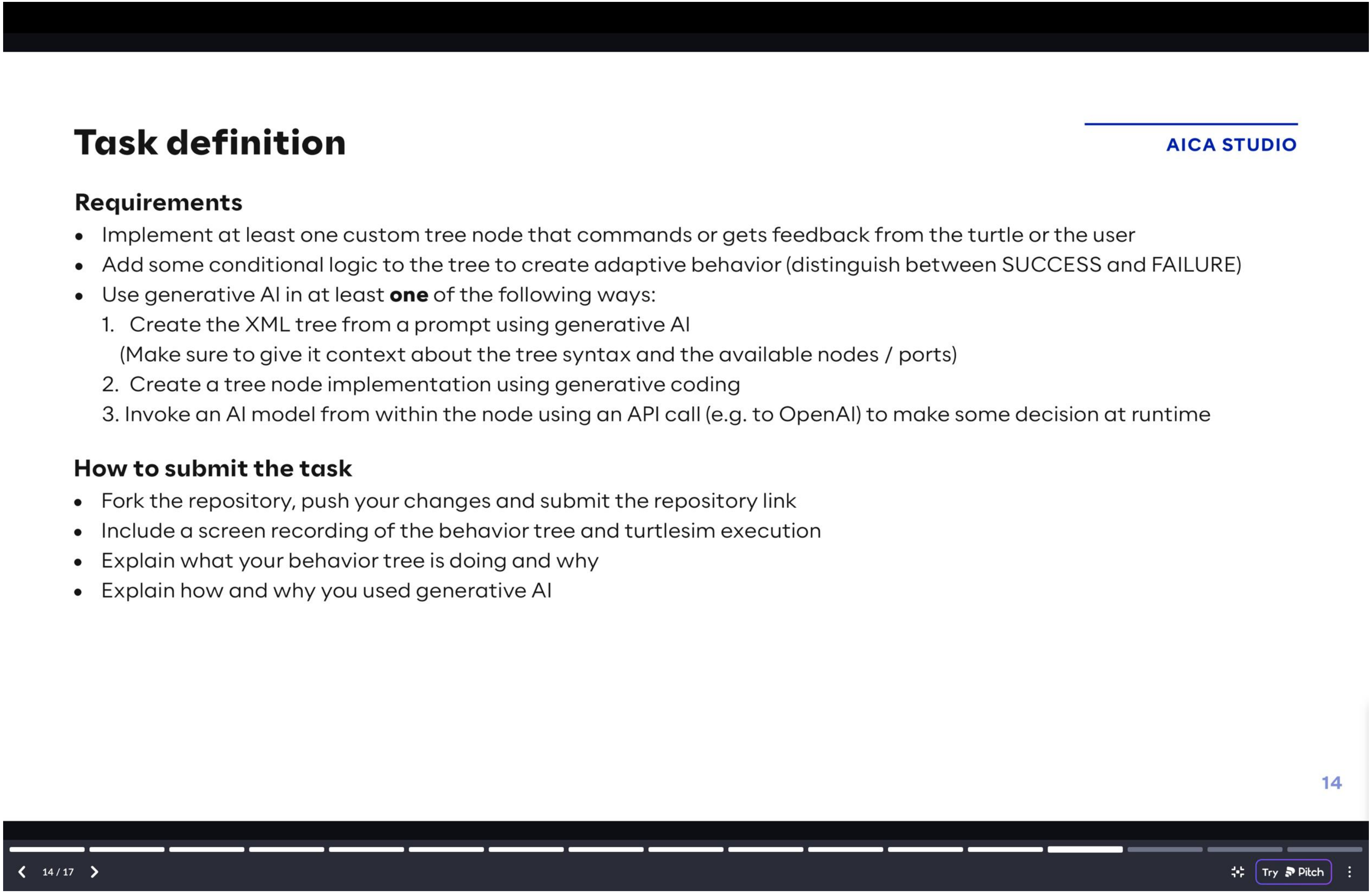
No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● C++ 89.3% ● Dockerfile 6.0%
● CMake 4.7%



Task definition

AICA STUDIO

Requirements

- Implement at least one custom tree node that commands or gets feedback from the turtle or the user
- Add some conditional logic to the tree to create adaptive behavior (distinguish between SUCCESS and FAILURE)
- Use generative AI in at least **one** of the following ways:
  1. Create the XML tree from a prompt using generative AI
     (Make sure to give it context about the tree syntax and the available nodes / ports)
  2. Create a tree node implementation using generative coding
  3. Invoke an AI model from within the node using an API call (e.g. to OpenAI) to make some decision at runtime

How to submit the task

- Fork the repository, push your changes and submit the repository link
- Include a screen recording of the behavior tree and turtlesim execution
- Explain what your behavior tree is doing and why
- Explain how and why you used generative AI