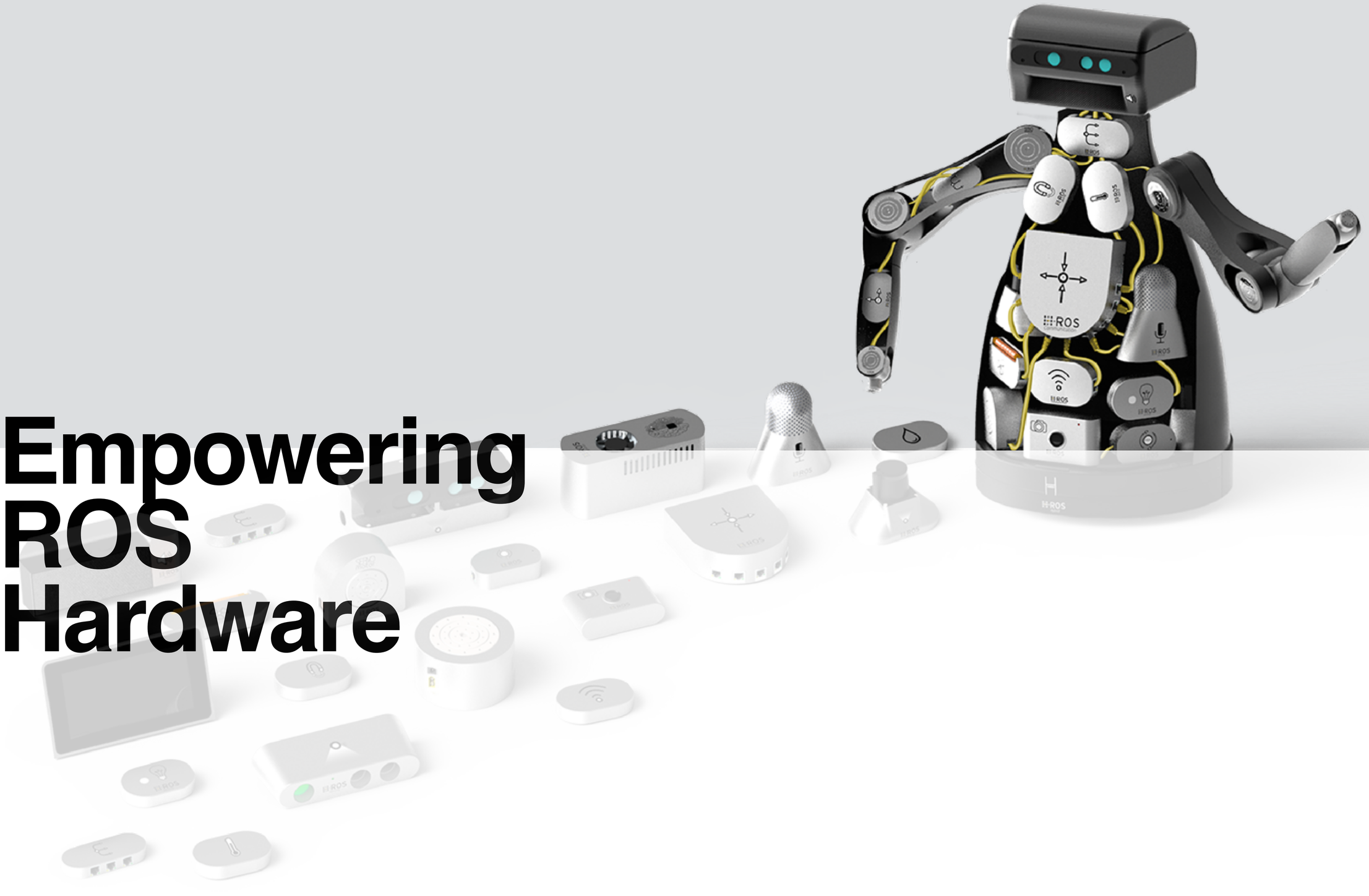
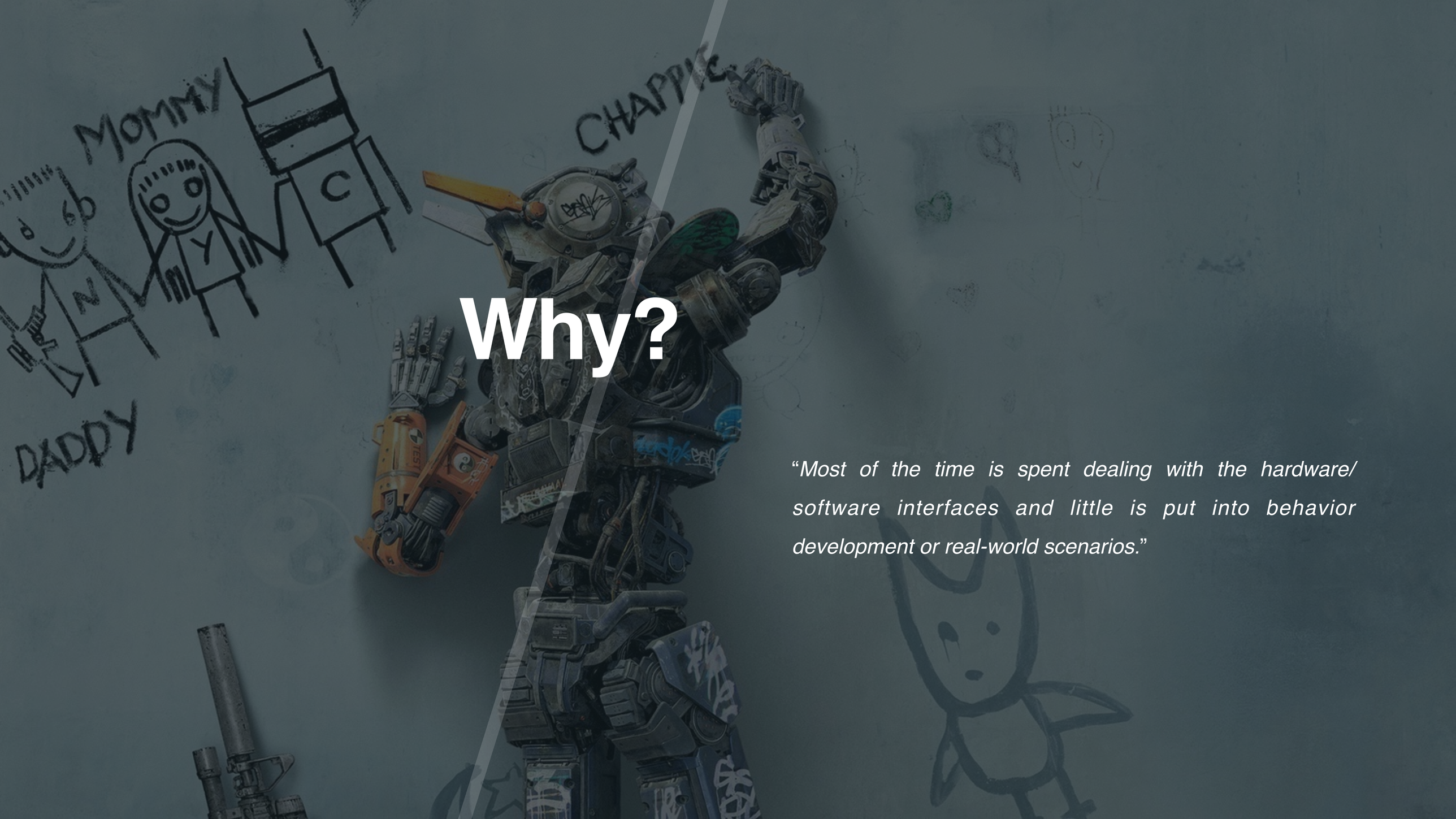
Hardware Robot Operating System

Víctor Mayoral Vilches
CTO and Founder, Erle Robotics

The Erle Robotics path

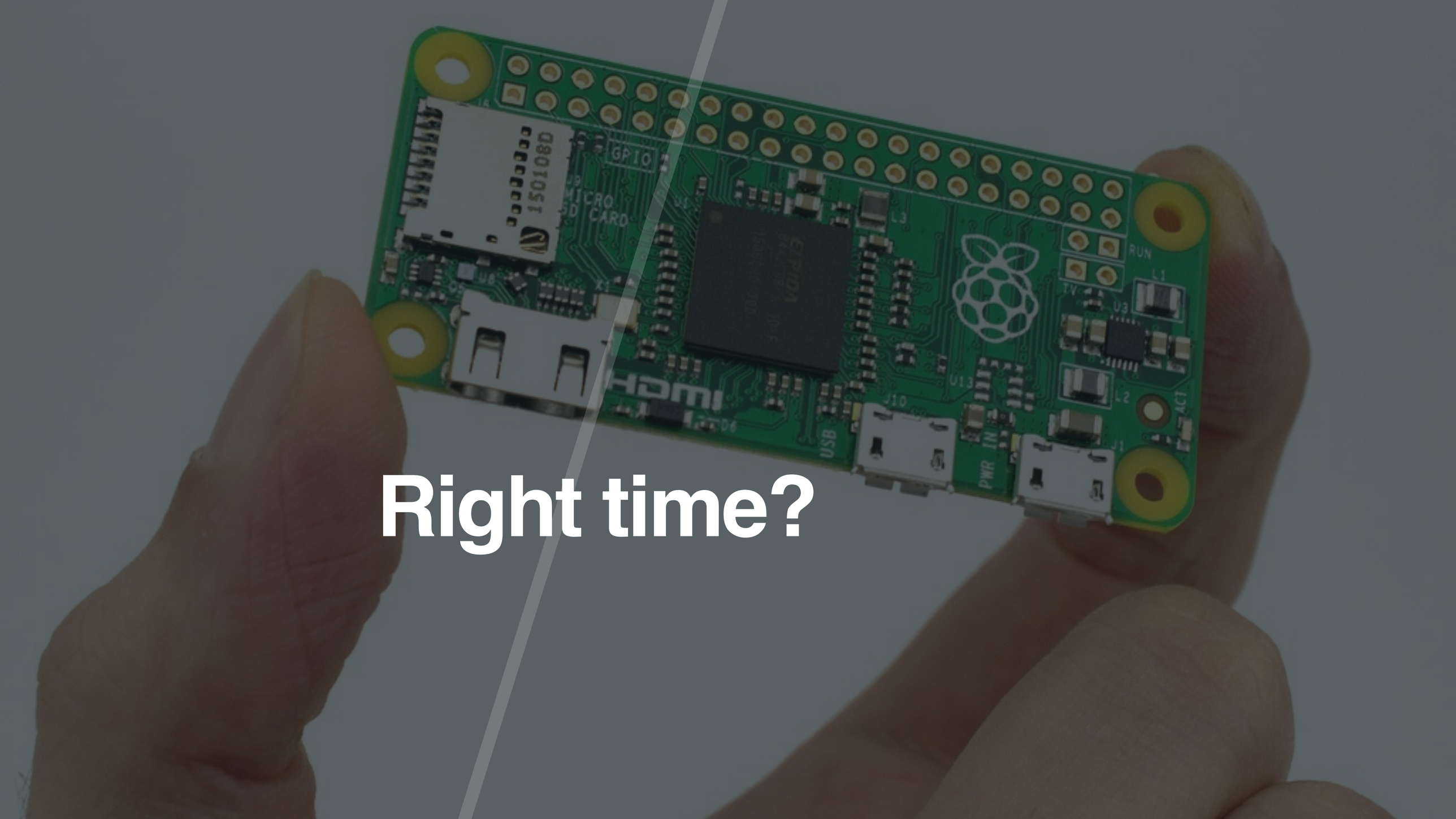
Empowering ROS Hardware



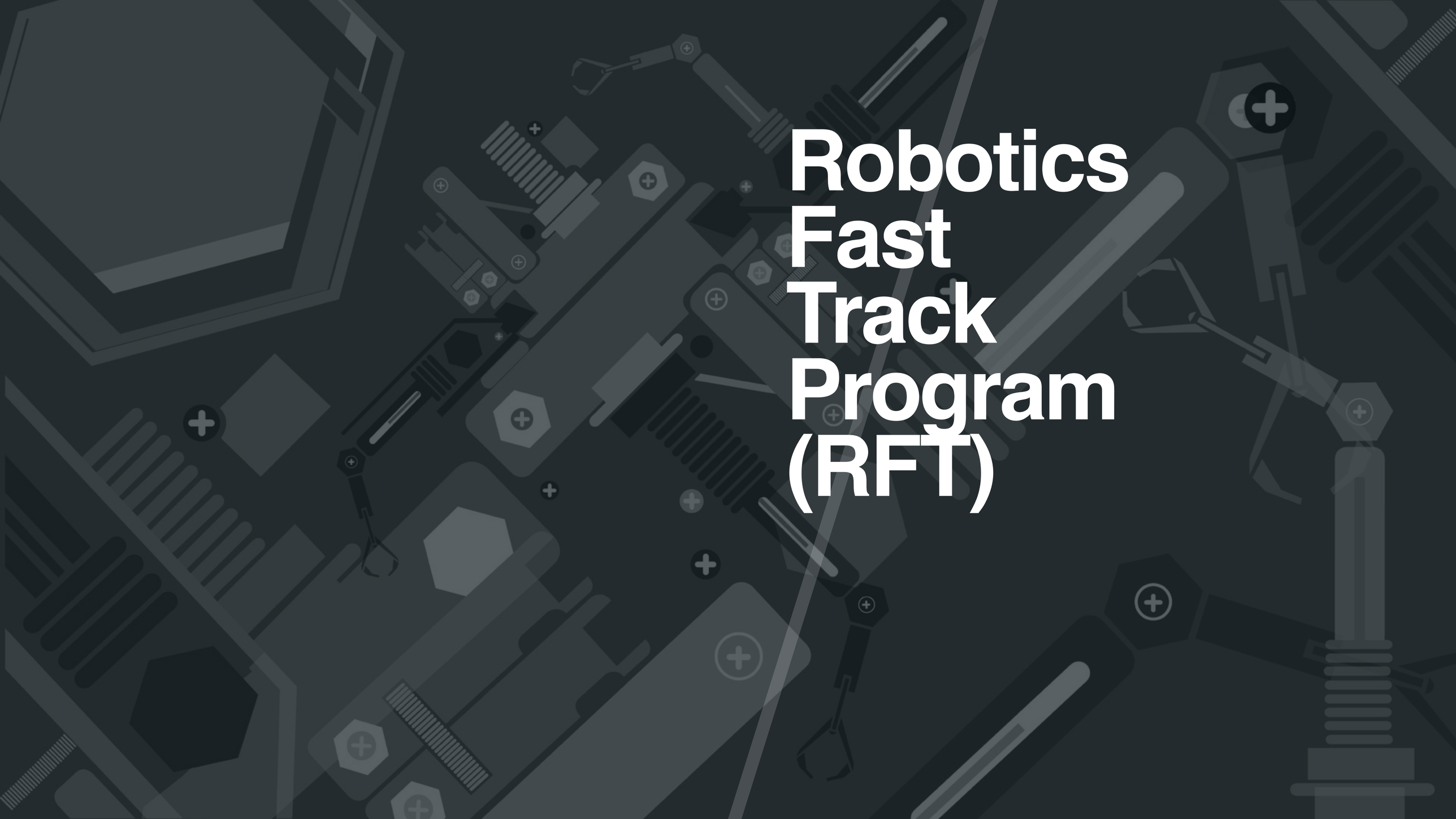


Why?

“Most of the time is spent dealing with the hardware/ software interfaces and little is put into behavior development or real-world scenarios.”



Right time?



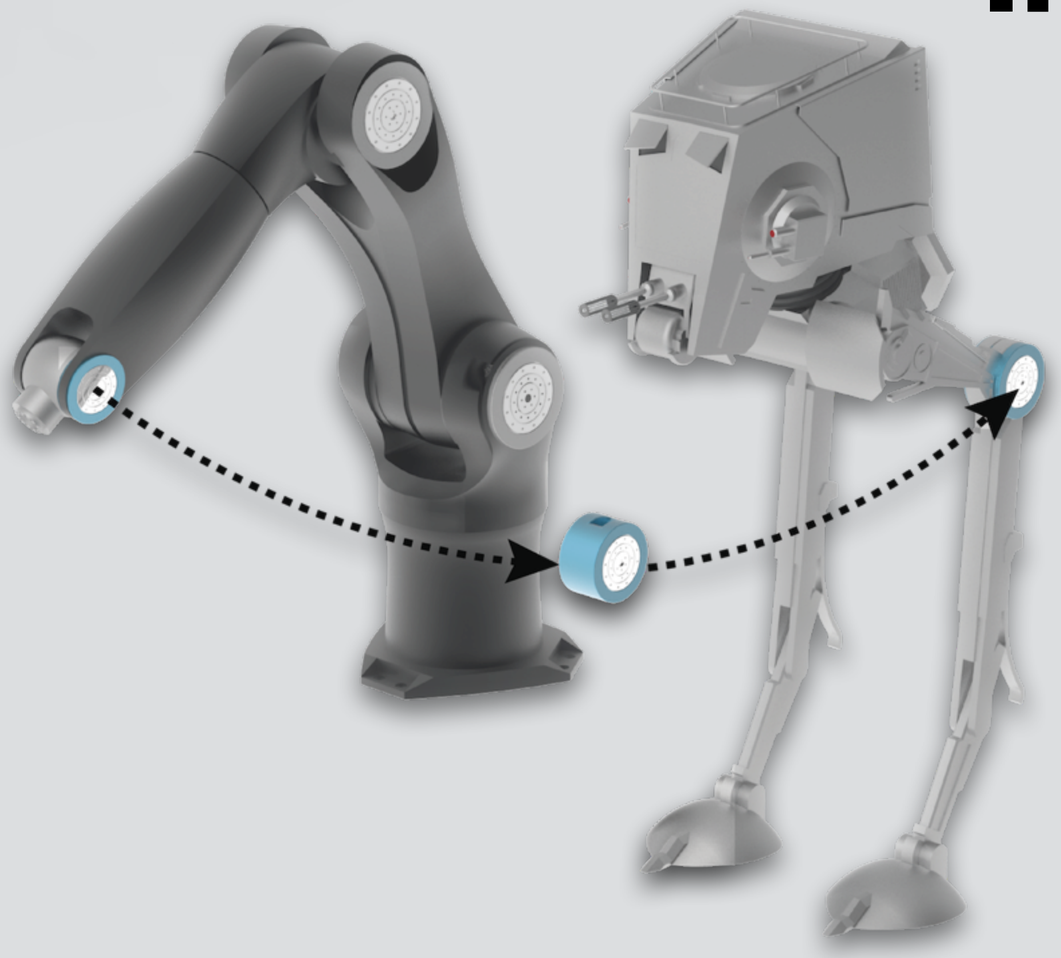
Robotics Fast Track Program (RFT)

Hardware Robot Operating System

A standardized software and hardware infrastructure to easily create reusable and reconfigurable robot hardware parts.

Using H-ROS, building robots is about placing H-ROS-compatible components together to build new robot configurations. The interesting fact about this is that constructing robots is not any more restricted to a few high technical skills but it's extended to a great majority with a general understanding of the different H-ROS parts and its use.

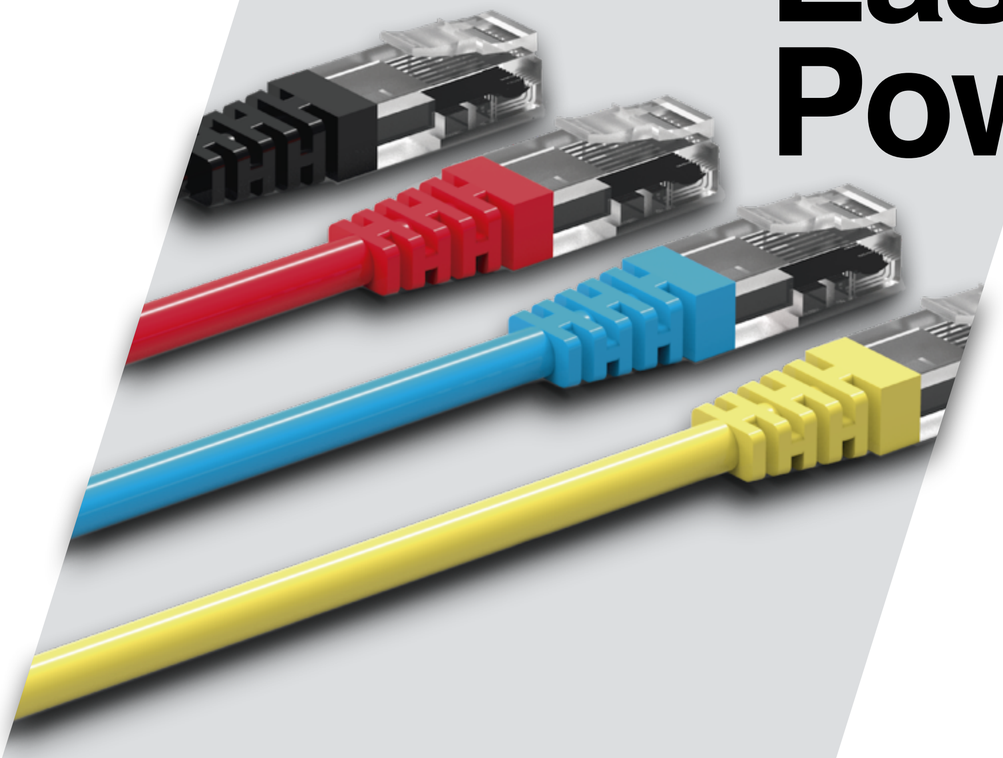
Interoperable



AT-ST

By facilitating standardized abstractions based in ROS, H-ROS compatible components are able to communicate and exchange data seamlessly, regardless of the manufacturer. This will help creating an environment of *compatible robot components* where parts from different manufacturers are literally interchangeable.

Simple.
Easy.
Powerful.

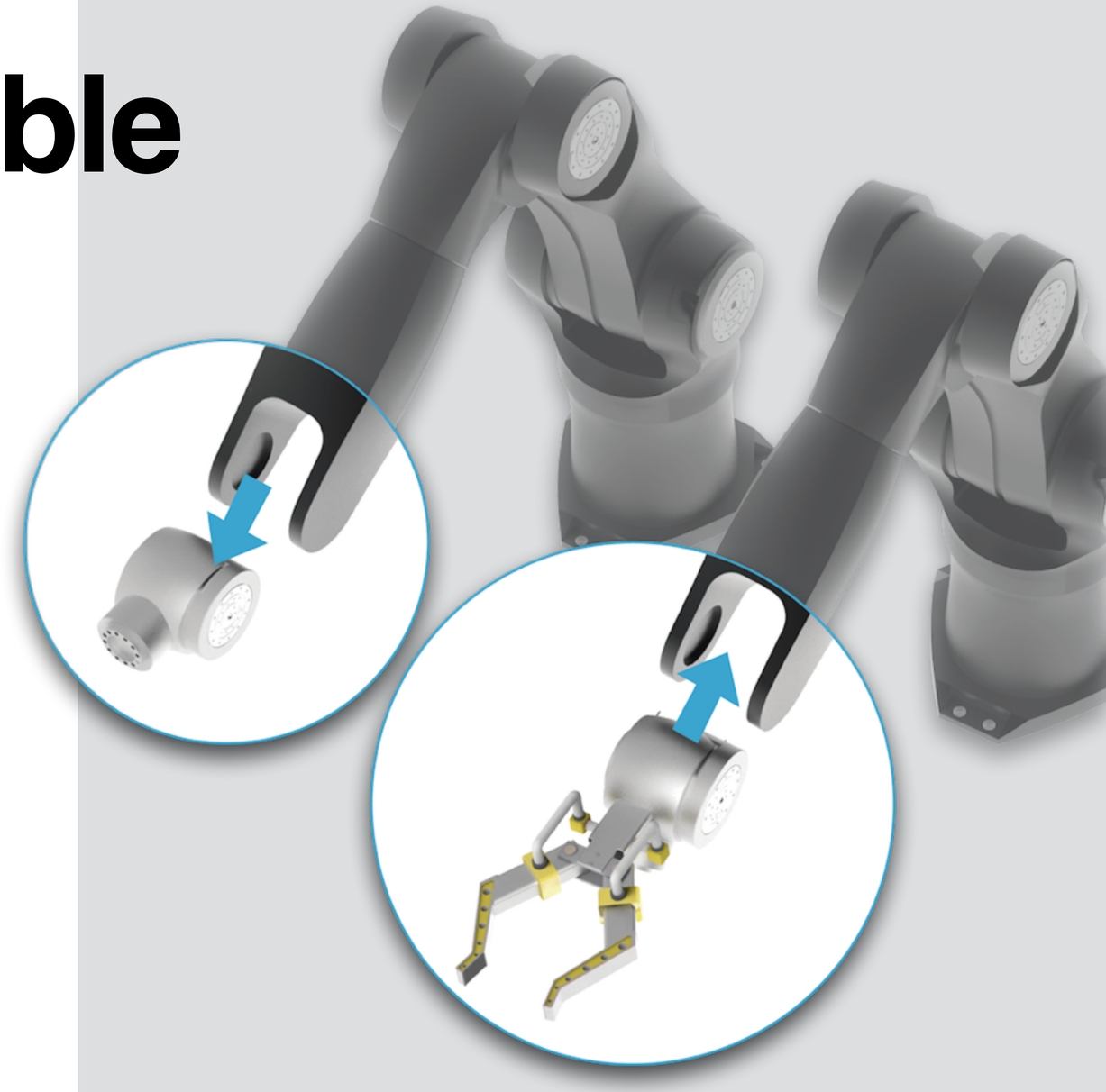


Power over Ethernet (PoE), *IEEE 802.3at-2009* (also known as PoE+). Particularly, mode B which delivers power on the spare pairs of 100BASE-TX.

This configuration provides 25.5 W of power to each H-ROS part connected through PoE using the RJ45 connector.

Reconfigurable

Our team extended the existing Unified Robot Description Format (URDF) to support dynamic changes and recreate the internal robot model based on a predefined set of routines.



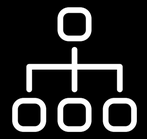
Built with ROS 2.0

The future framework for robot application development. Following from ROS, ROS 2.0 will define the next decade in robotics by providing solutions for problems like teams of multiple robots, small embedded platforms, real-time systems, non-ideal networks or production environments.

Standard Robot Components

Plug and play

Combine H-ROS components together to build new robots. Create, extend and repair robots easily.



Interoperable and distributed

Distributed hardware components that speak to each other regardless of the manufacturer.

Smart

Each components reports information that helps our robots become smarter (inertial position, voltage, current, …)

Blocks for robots

sensing

actuation

communication

cognition

hybrid

Sensing components for robots

H-ROS sensing components help robots perceive its environment and share information with the rest of the parts through standardized ROS interfaces.

Actuation components for robots

H-ROS actuation components allow robots to interact with its environment and produce some form of change through subscription and/or connection to standardized ROS interfaces.

Communication components for robots

H-ROS communication components are specialized in communication by either exposing new communication channels to the overall ROS network (e.g.: 4G, WiFi) or by providing means of interconnection between different H-ROS components.

Cognition components for robots

Cognition H-ROS components are specialized in computation and coordination. These parts perform most of the computationally expensive tasks within the robot such as reasoning, planning and/or reconfiguration.

Hybrid components for robots

Hybrid H-ROS components are composed of different sub-elements, that are generally not fully understood within any of the other types and/or correspond with sub-robots (parts of other robots). These components are grouped under an abstraction layer that enables them to interoperate directly with other H-ROS devices by complying with existing ROS abstractions.

The H-ROS workflow

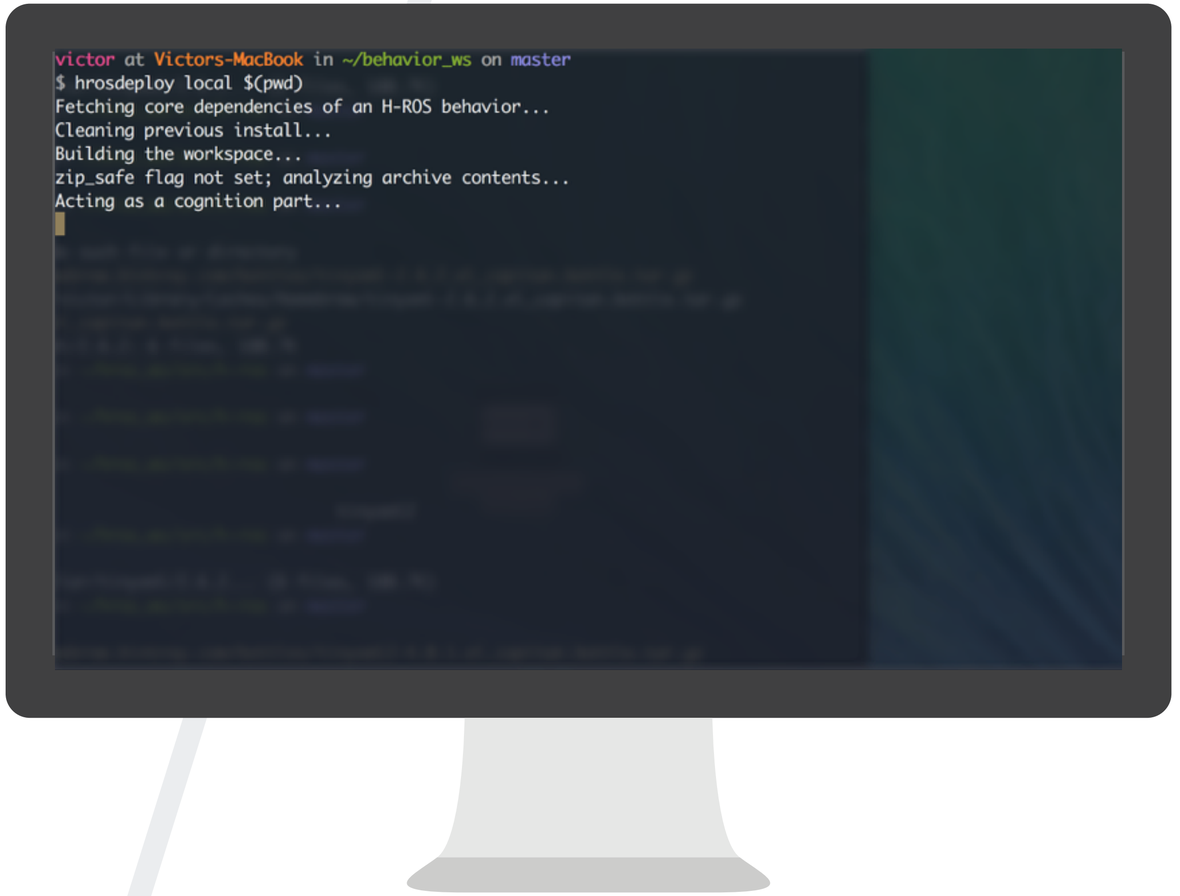
Step 1: pick your H-ROS components

Select the H-ROS components necessary to build your robot regardless of their size or manufacturer. They should all interoperate.

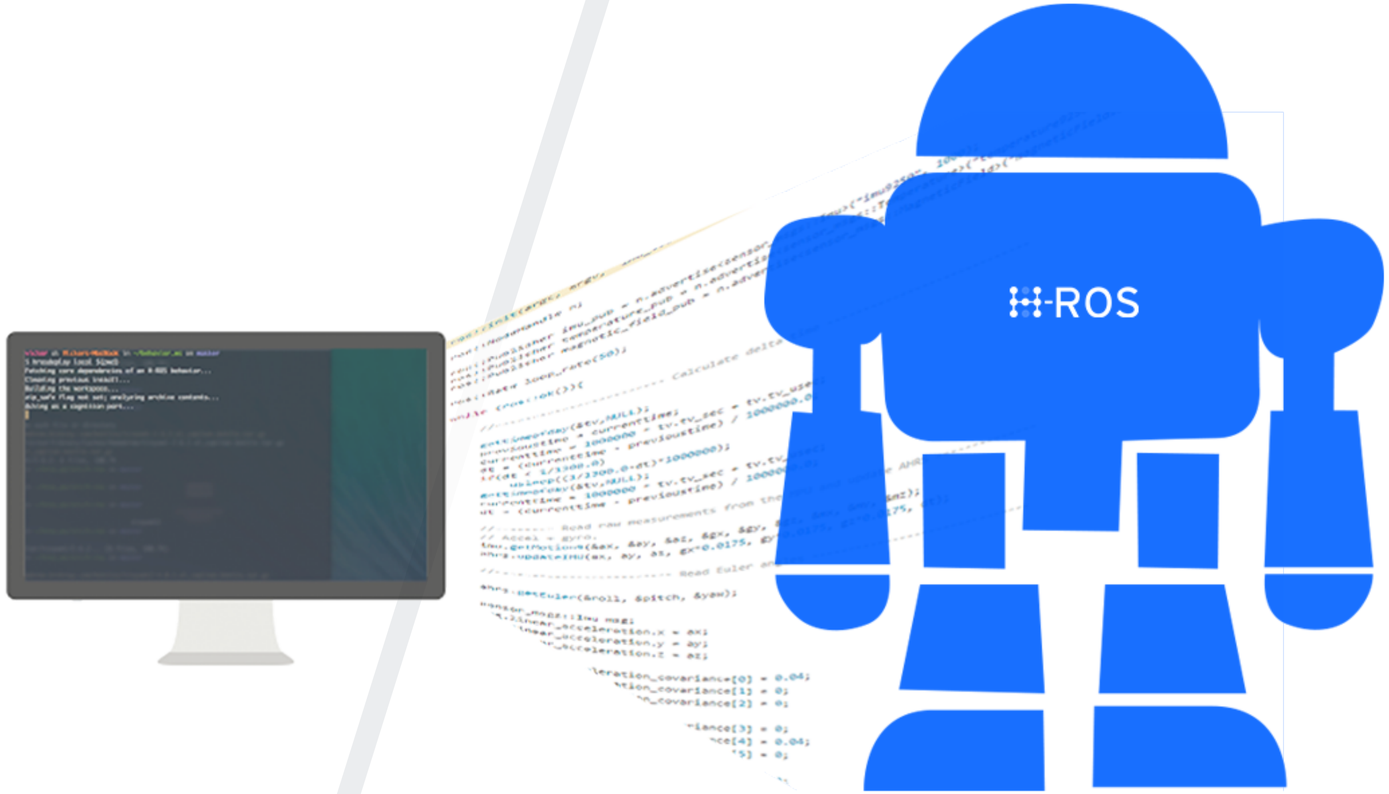
Step 2: code with ROS

We don’t want you to learn a new framework. Just use the tools that you know best and code amazing behaviour with ROS nodes, topics, and the rest of the abstractions.

Step 3: deploy and test locally



Step 4: ship the intelligence to the robot



H-ROS Turtlebot

Call for manufacturers

감사습니다
(thank you)

http://h-ros.com