MAGAZINO

Picked by a robot

Behavior Trees for real world robotic applications in logistics

Magazino GmbH
Landsberger Str. 234
80687 München
T +49-89-21552415-0
F +49-89-21552415-9
info@magazino.eu
www.magazino.eu

MAGAZINO

What is Magazino?

Flexible, mobile picking robots for your warehouse

MAGAZINO

Start-Up in the center of Munich

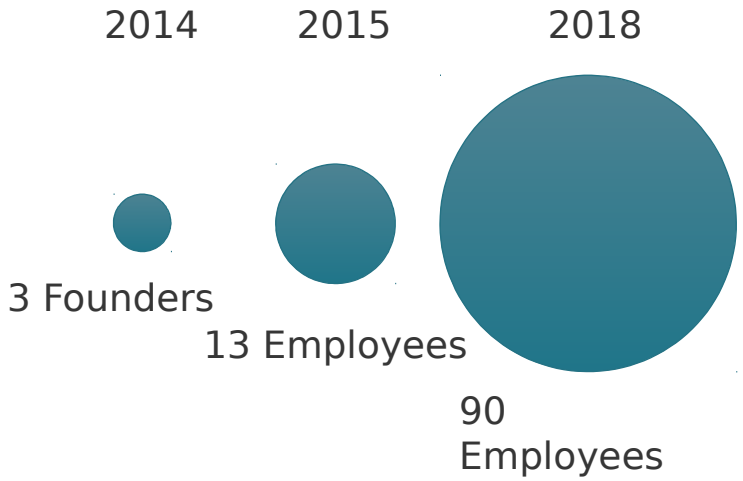
Completely automated order picking

Flexible robot solutions for intralogistics

Quick integration and scalation, as well as cost savings

Customers and partners

These companies trust Magazino



Worldwide first order-picking-robot in live-operation at one of our customers.

VDI Innovations Award 2017

2014 — Next47 Investment

2015 — 1. Customer
Robot TORU
Books

2016 — Development partner
Robot TORU
Shoe boxes

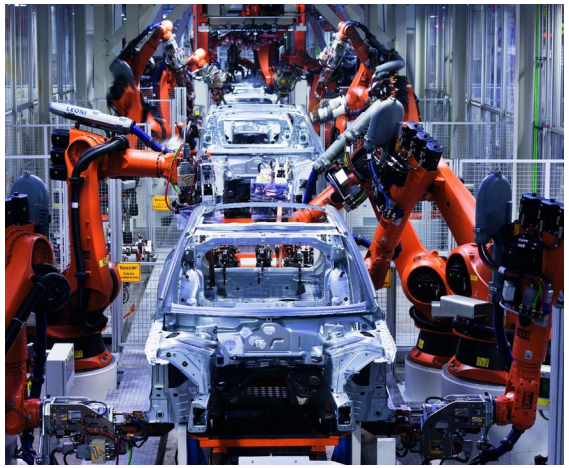
2017 — Development partner
3D vision system
Object recognition

2017 — Development partner
Robot SOTO
Cardboard boxes

2018 — $25M Investment by Körber, Zalando, Fiege and Cellcom

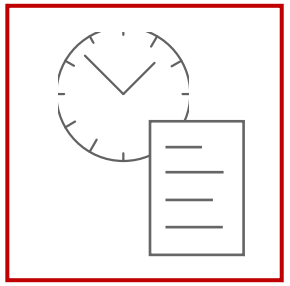
Problems in the fulfillment sector

Uncertainties set limits for automation

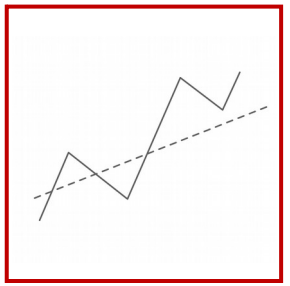


Highly automated and fast production at Audi today

Almost completely manual intralogistics in the same factory today



Short contract periods



Handling order peaks

Unergonomic tasks

Lack of qualified personnel

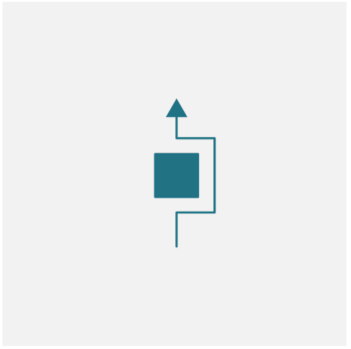
Robotics Technologies Comparison

A huge step compared to traditional robotics

MAGAZINO

+ High precision & performance
- Repetitive, predefined jobs
- Deterministic tasks

Sensor-based



Behavior adaption and decisions at runtime

Learning with artifical intelligence

Cloud-based

Approaches by the automation industry

Concepts at work right now

MAGAZINO

Concept	Goods-to-man	Man-to-goods
Products	 	 
Manufacturer	SSI SCHÄFER amazon robotics kardex remstar	KNAPP PHOENIX CONTACT PROGLOVE wearables for industry

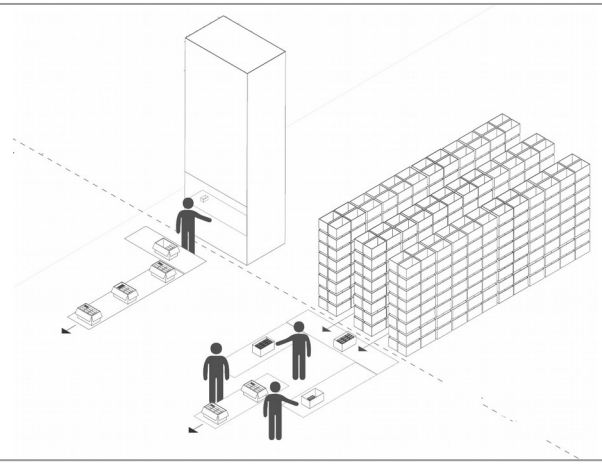
Rigid concepts

Why these concepts are not the final solution

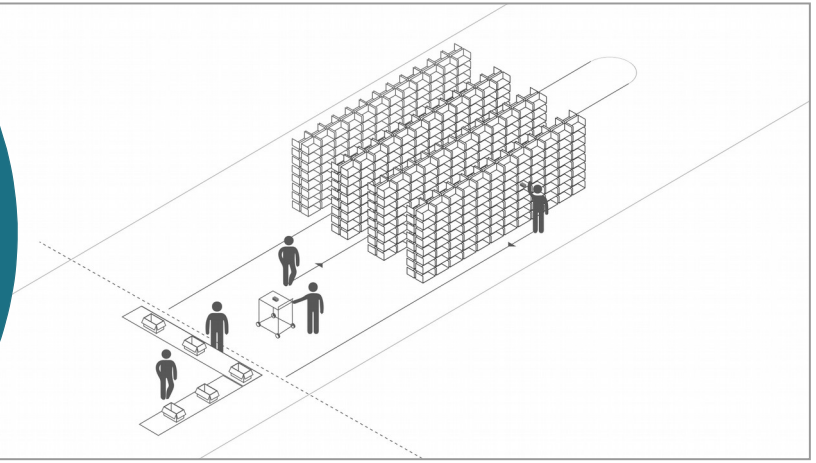
MAGAZINO

Main factor

Goods-to-man



Man-to-goods



- High initial investment
- No flexibility
- No scalability

Human

- Hardware concepts only as support for humans
- High employee costs

Cooperative robots for intralogistics

Magazino‘s solution for fulfillment and production supply

MAGAZINO

TORU

SOTO

MAGAZINO

Pick-by-Robot for the fulfillment sector

Supply-by-Robot for production lines

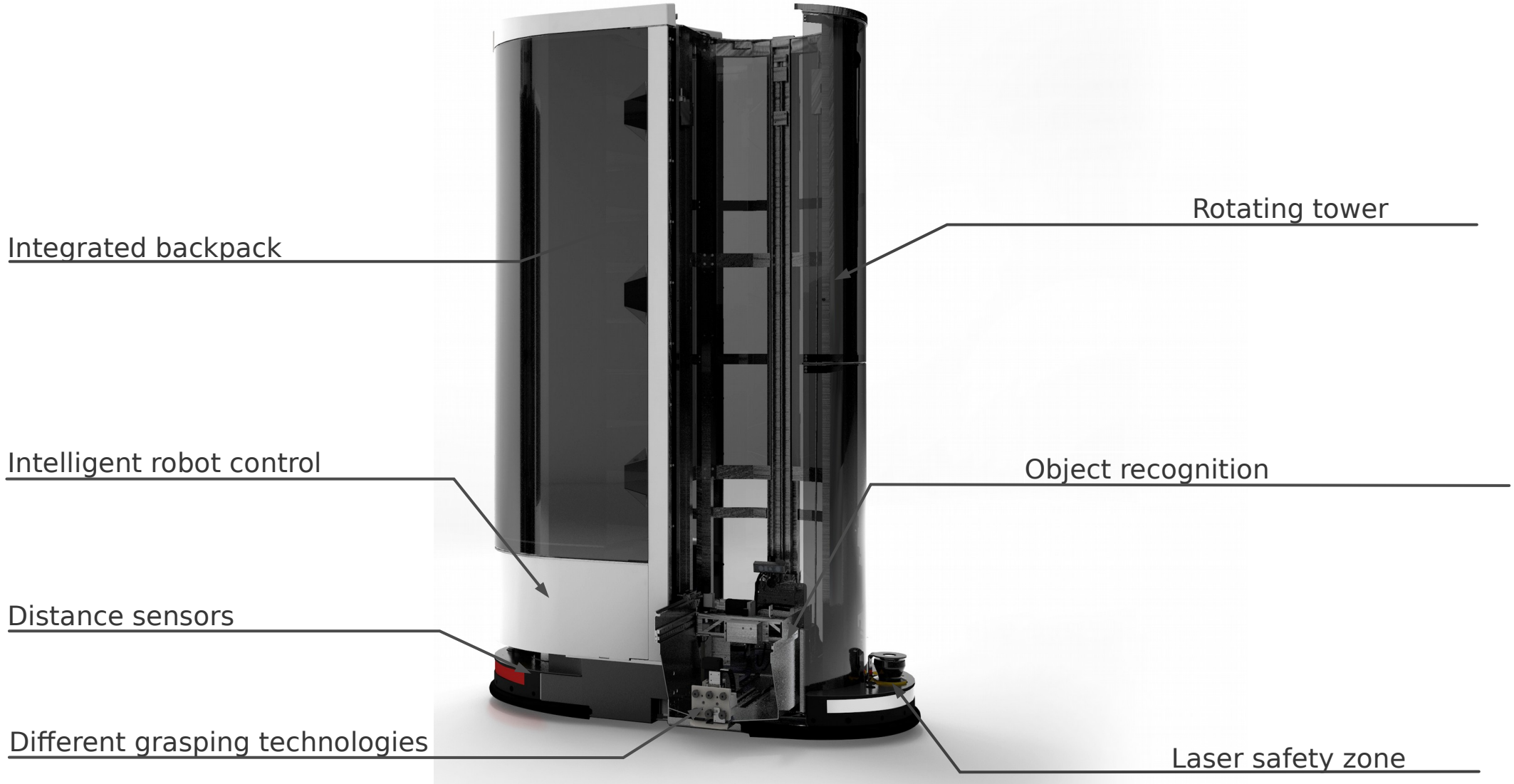
MAGAZINO

TORU Pick-by-Robot

TORU

The robot for the E-commerce sector

MAGAZINO

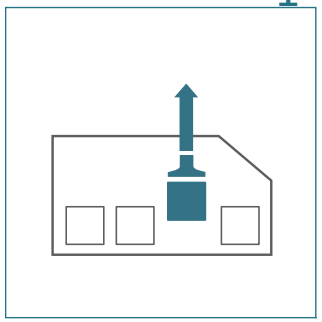


MAGAZINO

Advantages of TORU

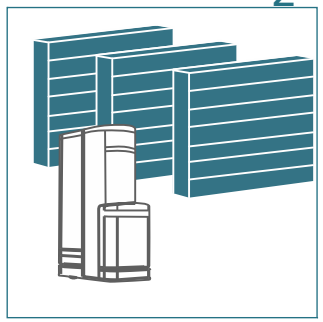
Flexible automation to save pick-costs

1



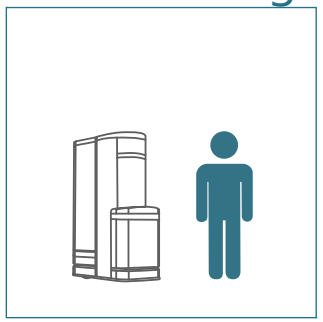
Item-specific handling

2



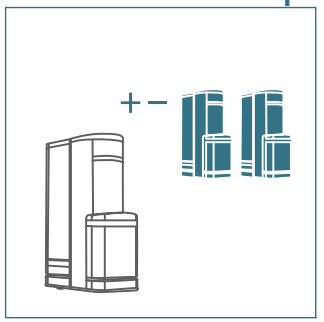
Integration into existing warehouse

3



Working parallel to humans

4



Flexibility

5

Reduced labor- and process-costs

6

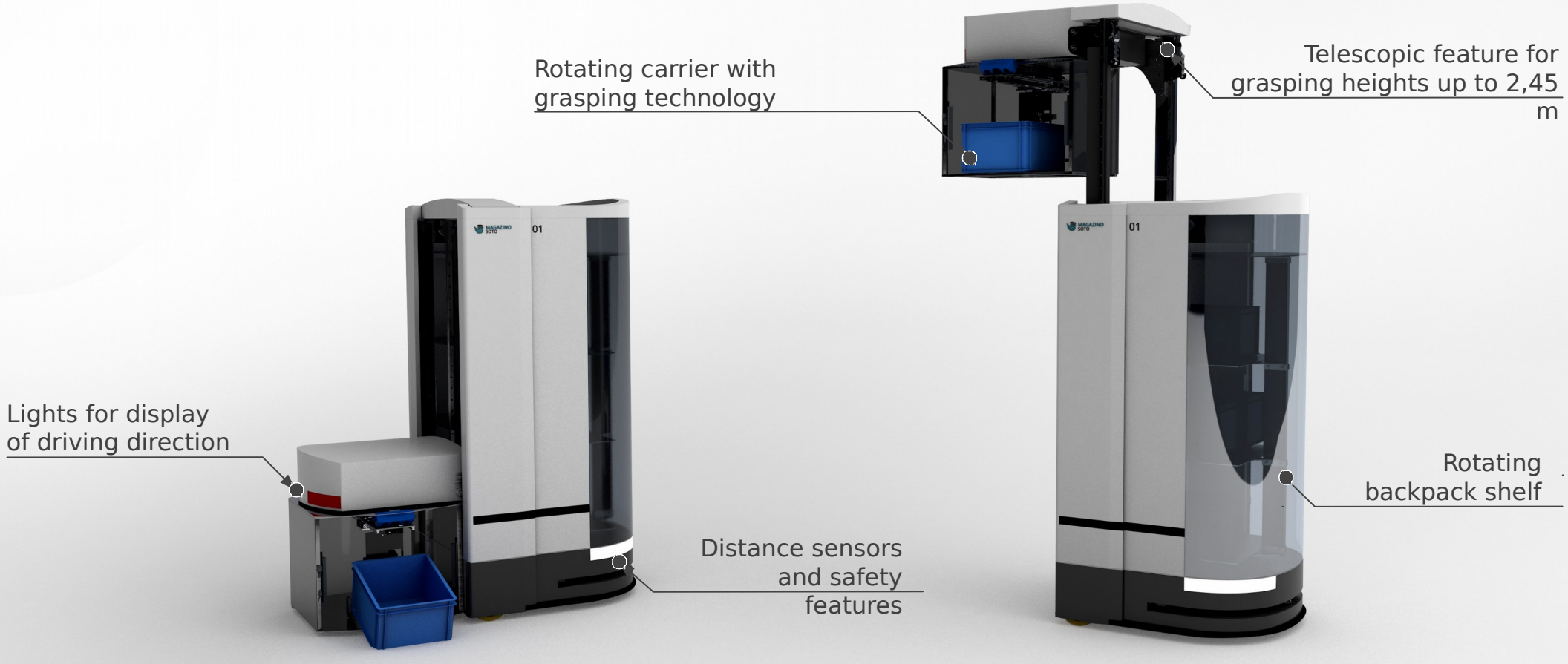
New logistics concepts possible

SOTO Supply-by-Robot

SOTO

The robot for production supply

MAGAZINO

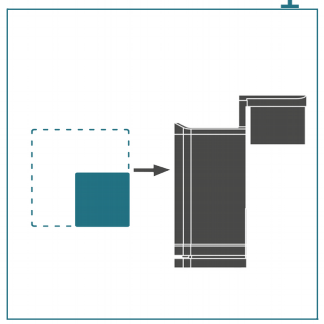



MAGAZINO

Advantages of SOTO

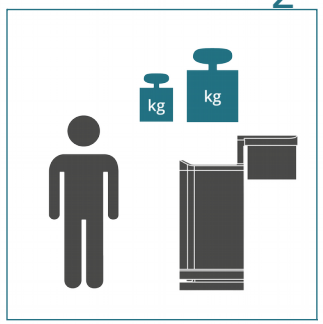
Saving costs through automated production supply

1



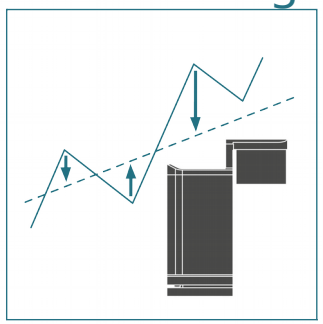
Flexibile gripping

2



Support for unergonomic tasks

3



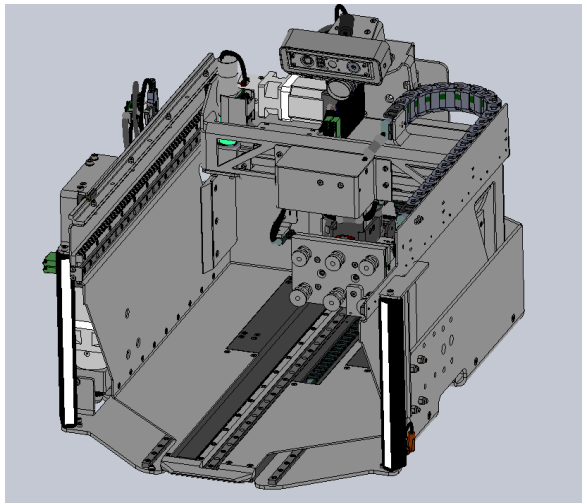
Lower stock because of Just-in-time delivery

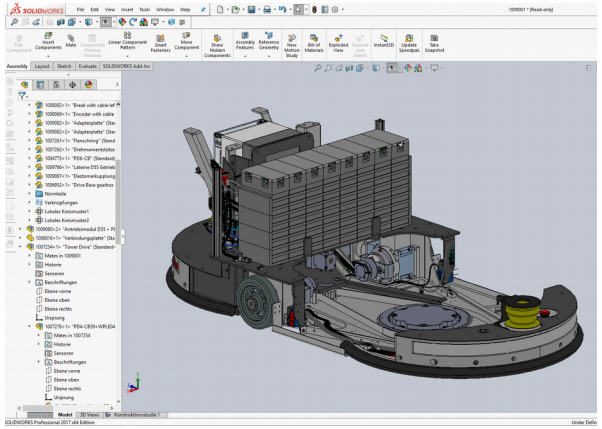
4

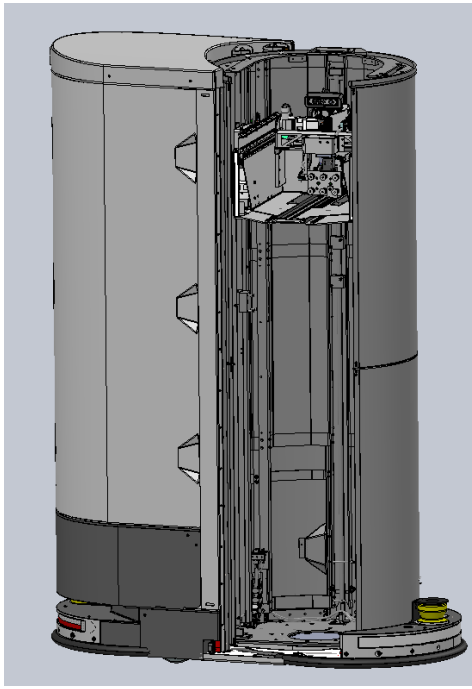
Reduced labor- and process-costs

Mechanical design

- The entire robot is designed internally by Magazino engineers

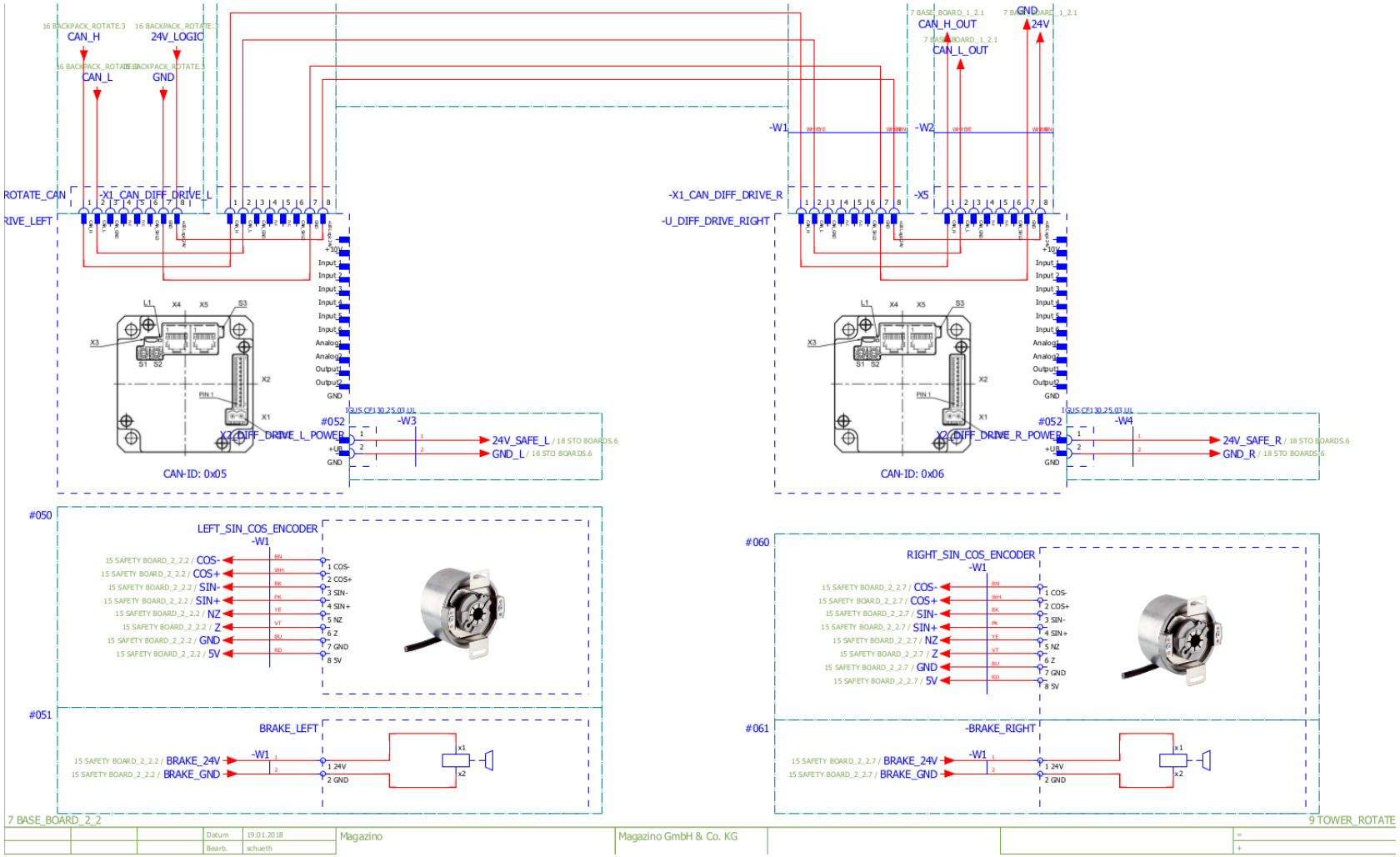






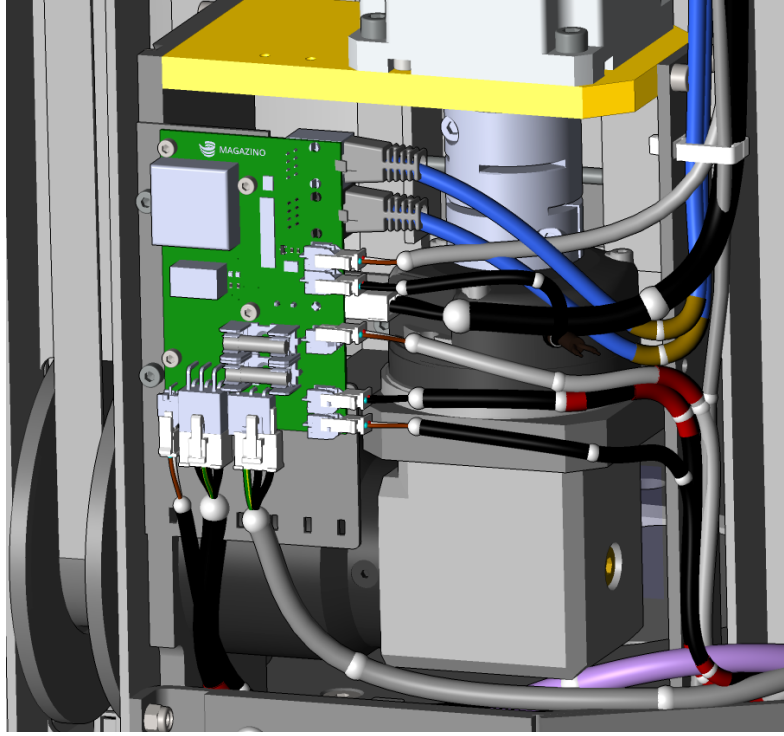
Electronic design

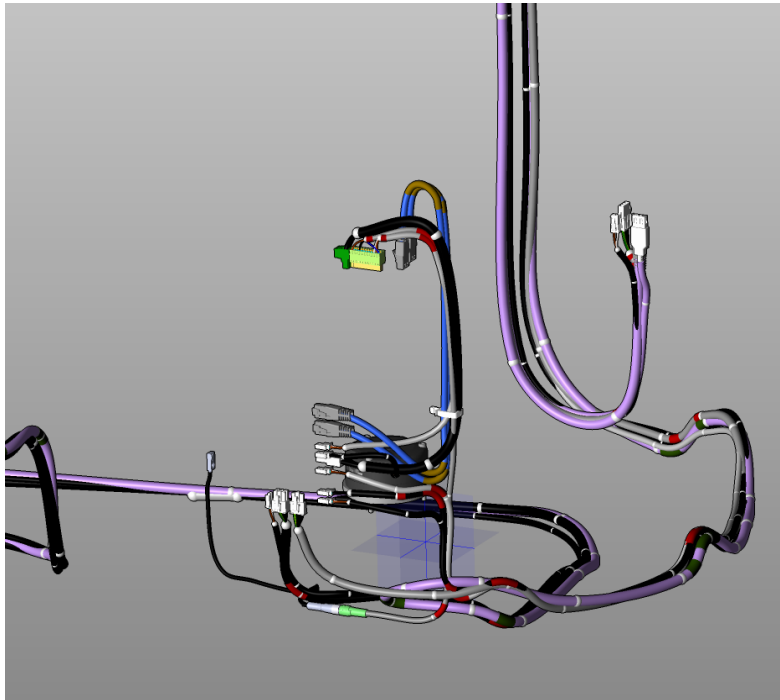
- Robot uses custom PCBs to dispatch power and interface with hardware



Electronic design

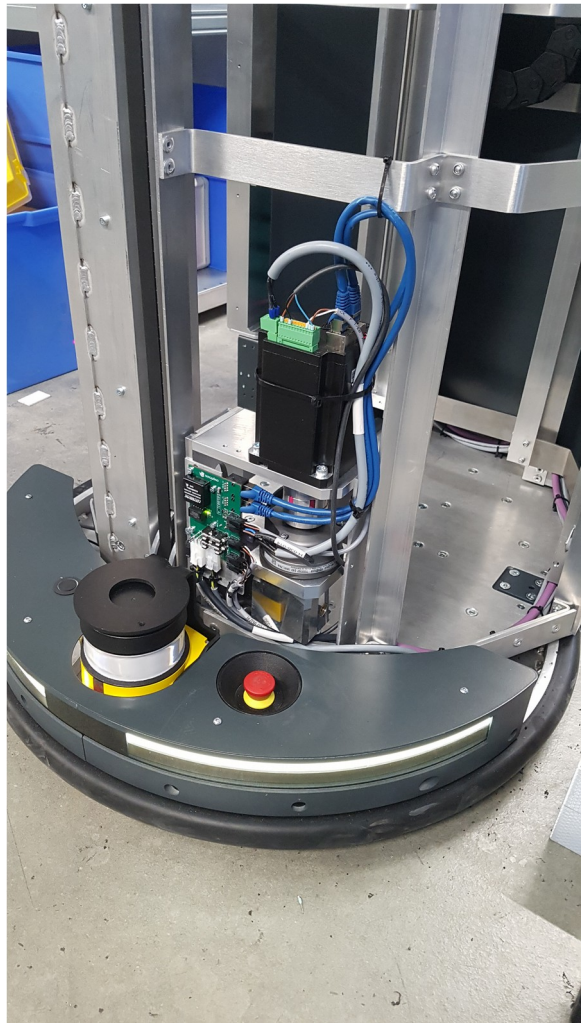
- The 3D mechanical drawing is augmented with all the cables
- Collisions, lengths and maximum bending are included
- PCB size and connectors placements are added in the CAD





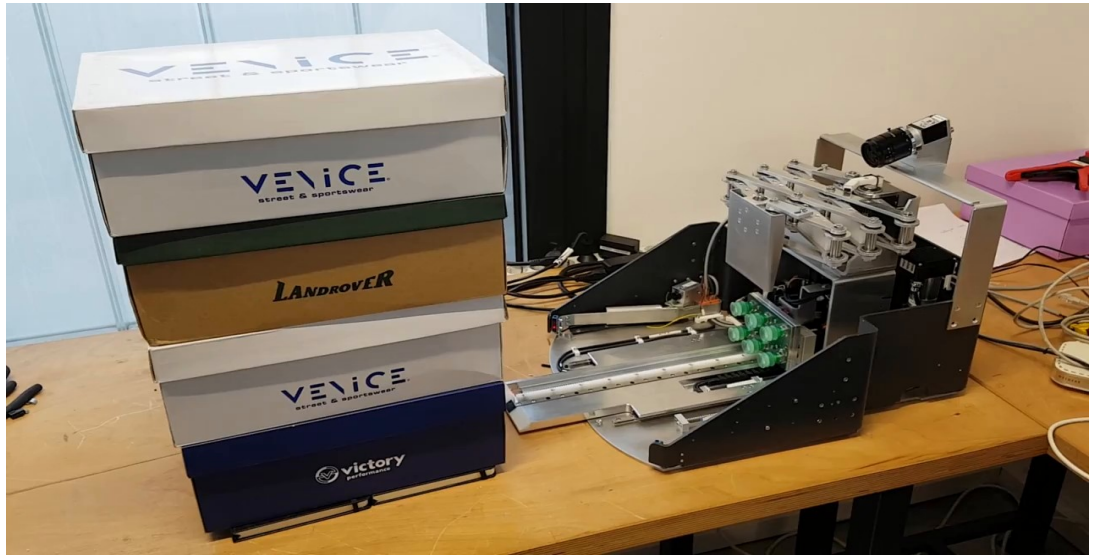
Magazino robots are built internally

We have a production department where robots are built and tested



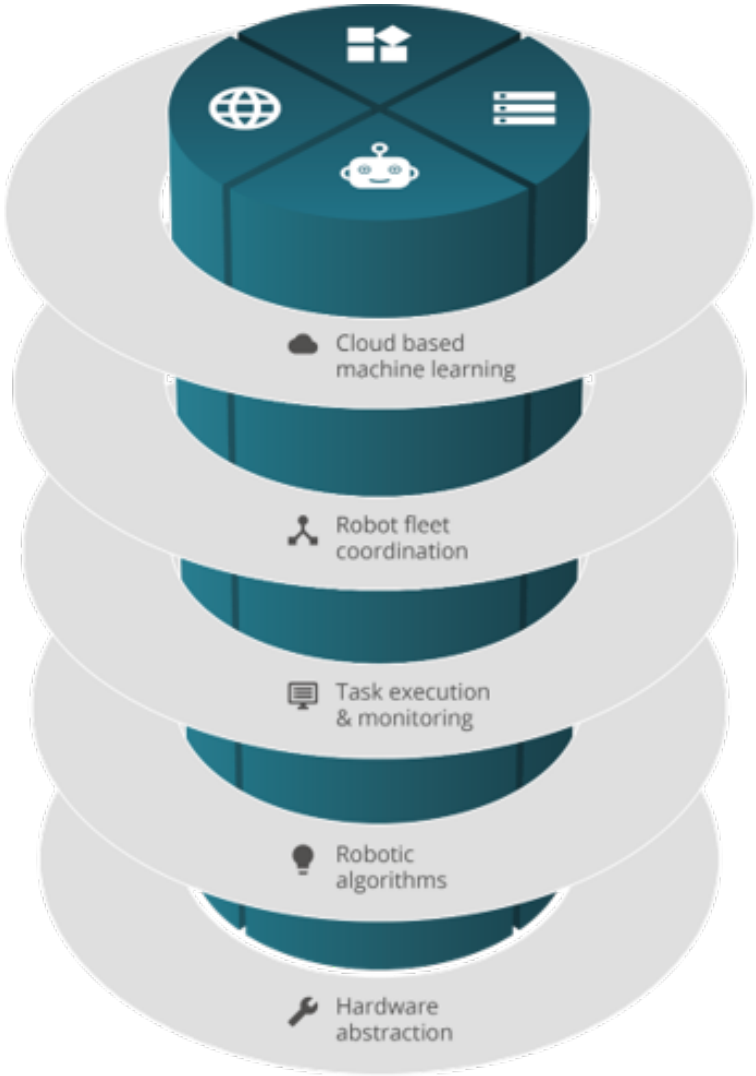
Magazino robots are built internally

We have a production department where robots are built and tested



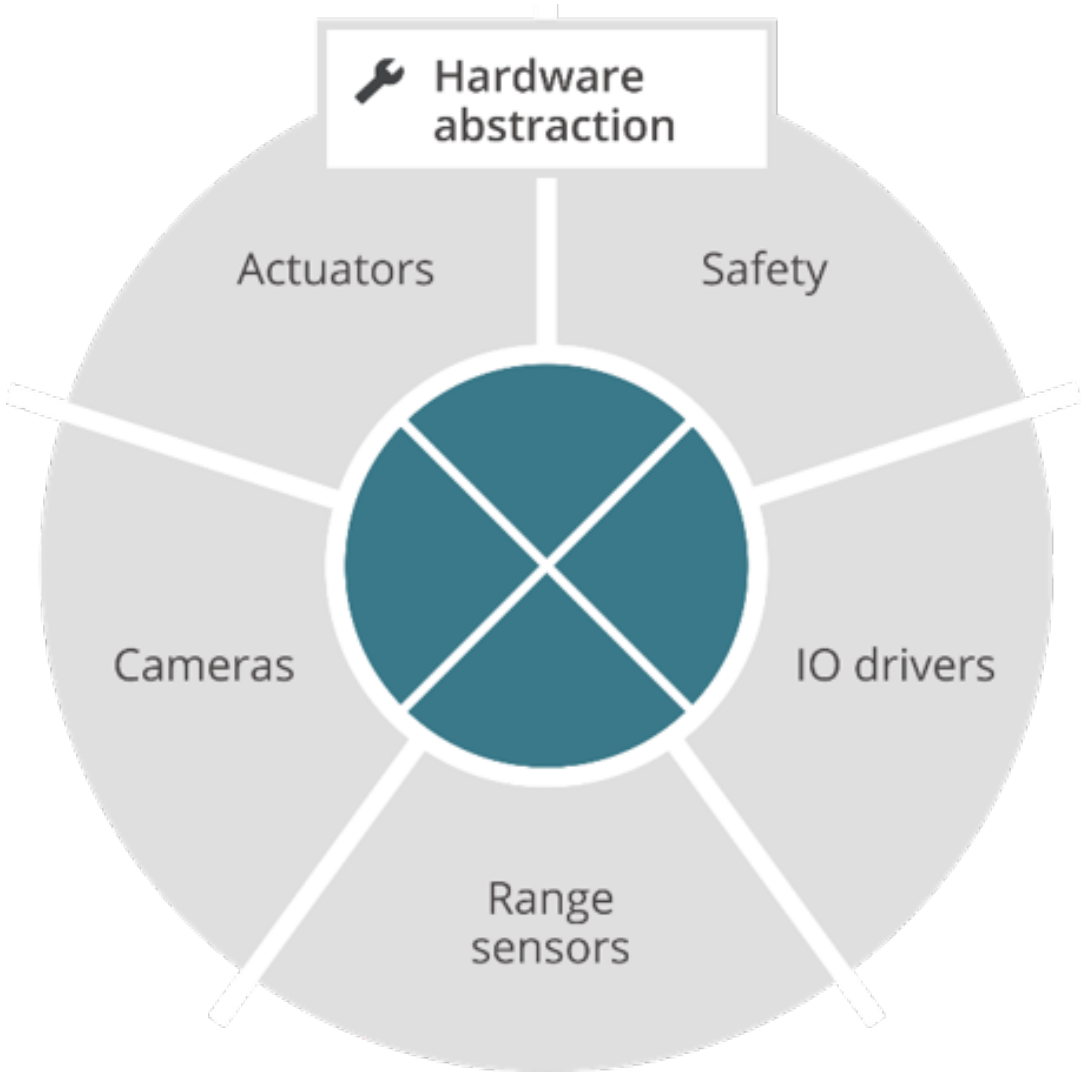
Software architecture (ACROS)

A unique framework architecture for perception-guided engineering that is generalizable to other robots and environments



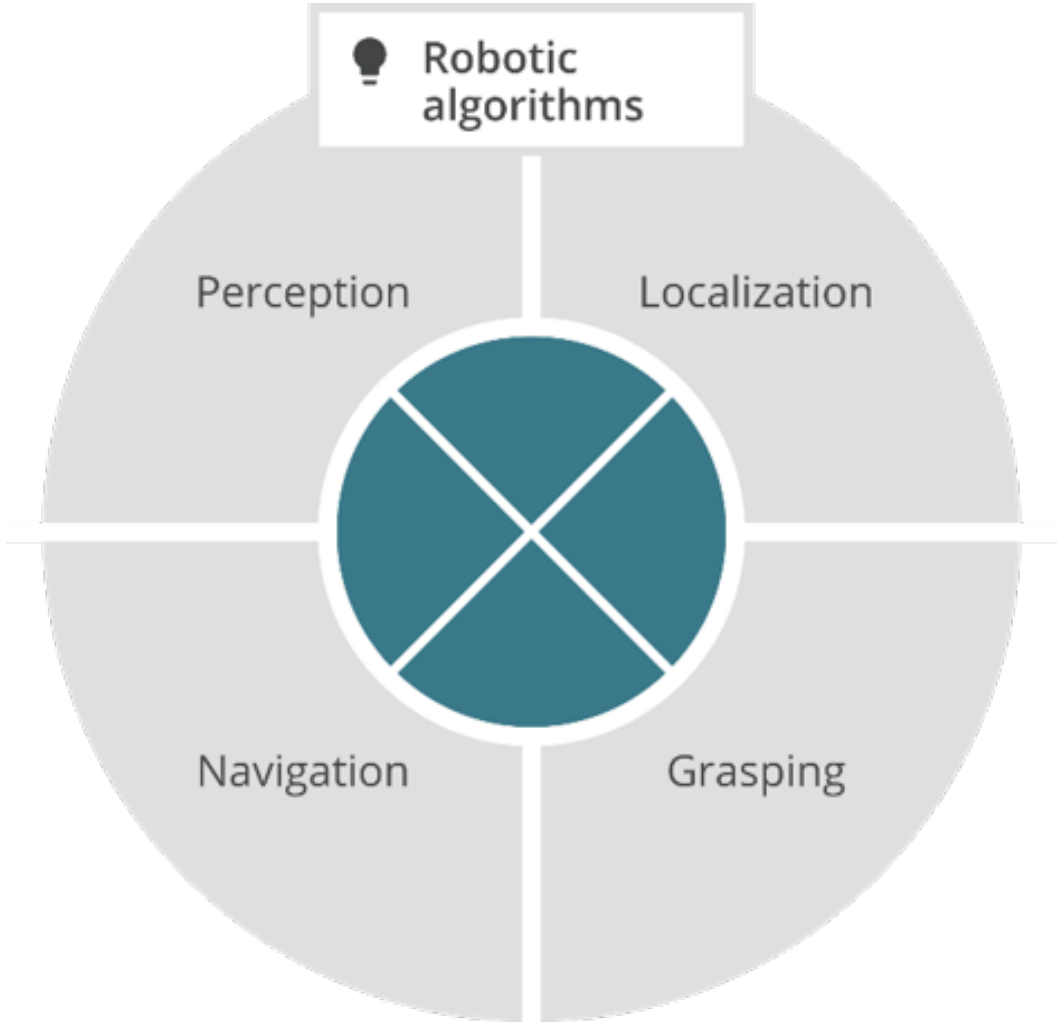
Hardware abstraction

A broad range of drivers decouples the upper software layers from the hardware, allowing the use of many different components



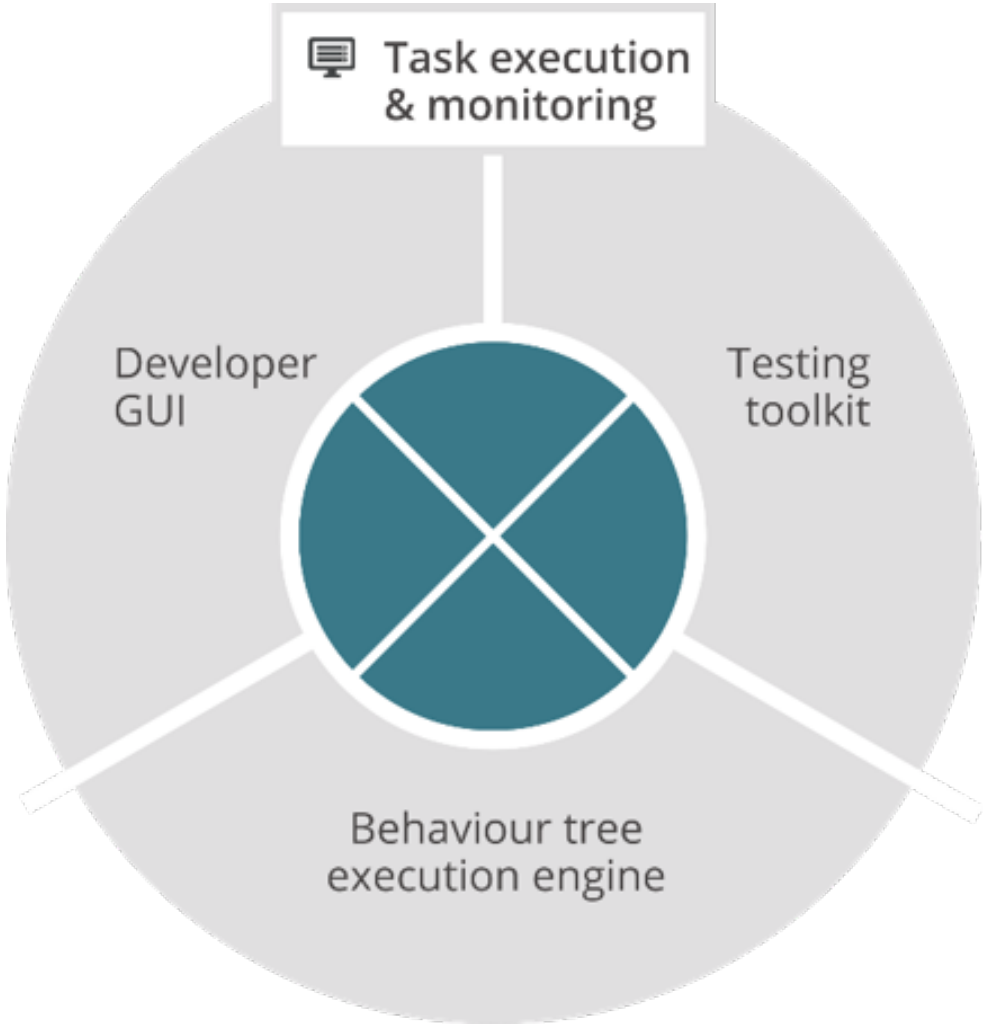
Robotics algorithms

A suite of algorithms for interpreting sensor data provides the components for engineering robot applications



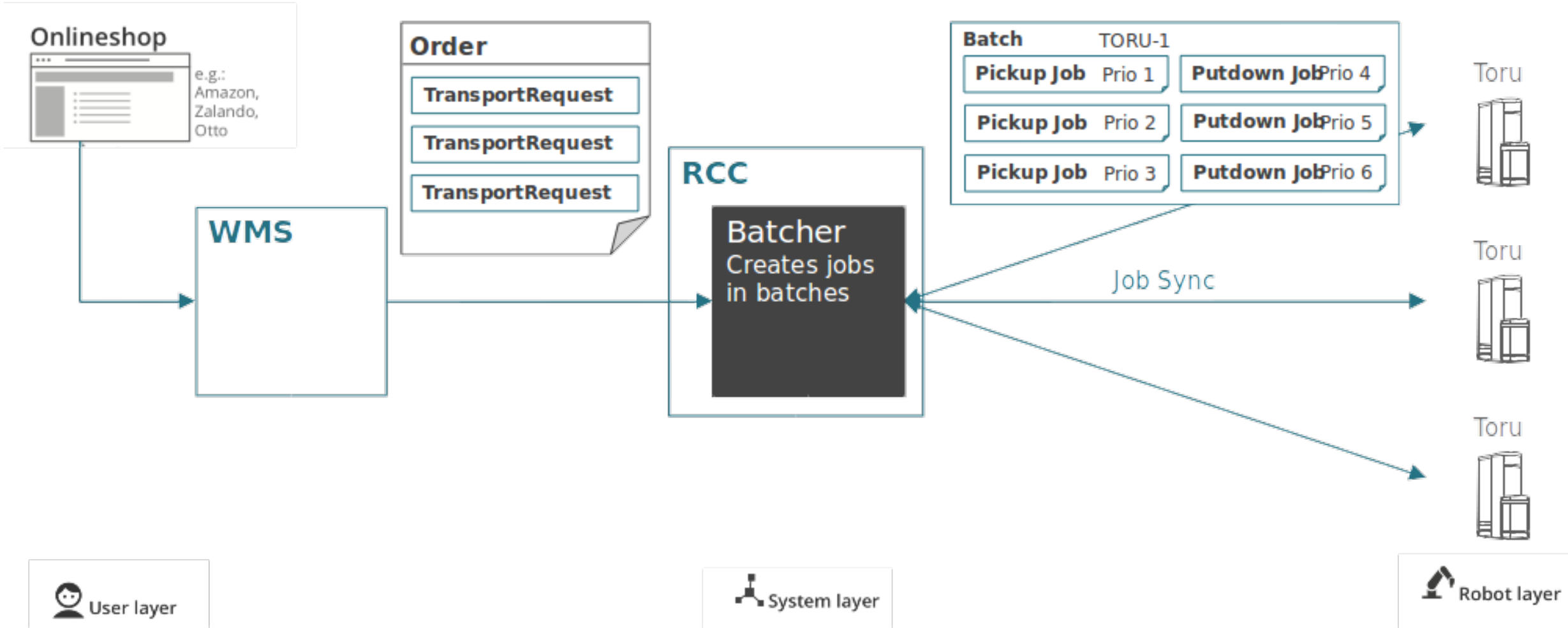
Task execution and monitoring

Behavior Trees support the efficient modeling, execution, and supervision of highly reactive perception-guided robot applications

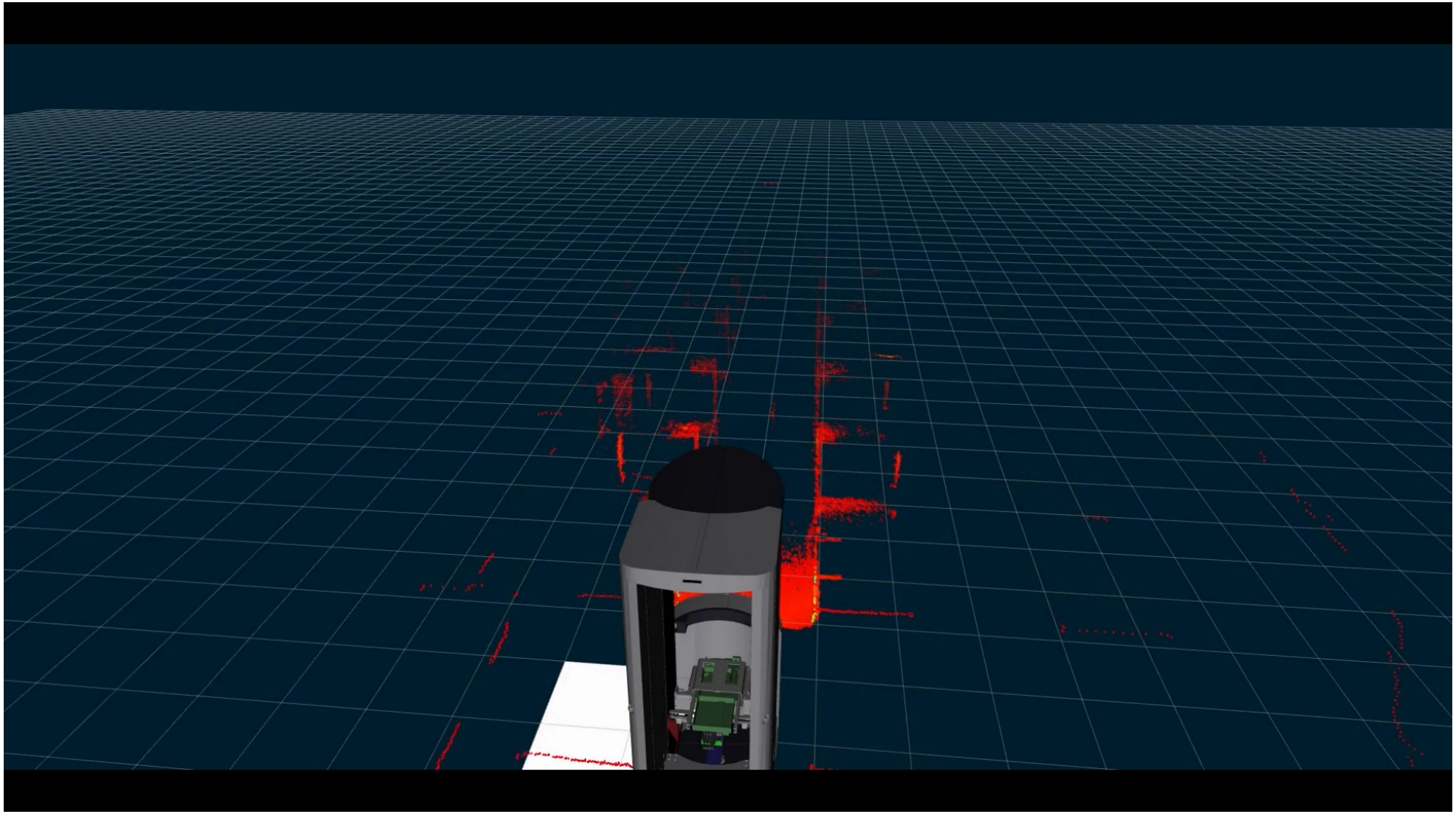


From WMS to robot

Robot Control Center (RCC) controls robots by task allocation and prioritization



Perception/Grasping: How does the robot see the world?

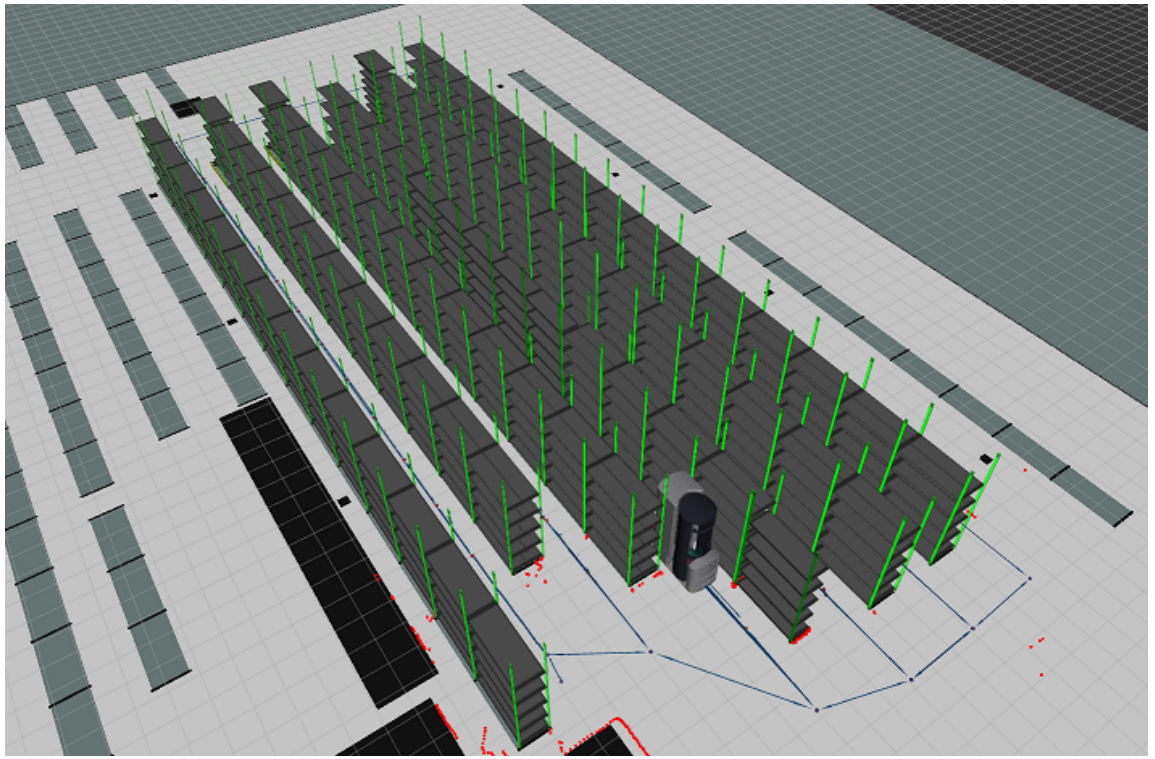


How to deal with the problem?

Behavior Trees for real world robotic applications in logistics

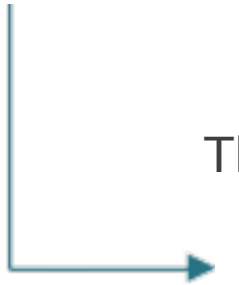
The Robotic Problem

- Robots work in warehouses where humans work
- Robots are requested to move objects around (move $\mathcal{I}$ from A to B)



A simple problem but...

- Enormous amount of situations to be faced
- Different customers to handle
- Complaints coming from customers
- Multiple type of robots



The system requires:

- Flexibility
- Adaptability
- Introspectability
- Reusability and scalability

We were looking for an executor able to cope with such requirements.

Behavior Trees

A Behavior Tree (BT) is a mathematical model of plan execution used in computer science, robotics, control systems and video games. They describe switchings between a finite set of tasks in a modular fashion. Their strength comes from their ability to create very complex tasks composed of simple tasks, without worrying how the simple tasks are implemented.

wikipedia.org

Origin of Behavior Trees and Literature

- BTs originate as tool to model the behavior of NPCs
- They have been used in games such as Halo, Bioshock, and Spore

- First paper in literature:
  - Handling Complexity in the Halo 2 AI, Isla D., GDC 2005

- In robotics:
  - Towards a Unified Behavior Trees Framework for Robot Control, Marzinotto et al., ICRA 2014
  - Controlling Process of Robots Having a Behavior Tree Architecture, Tenorth, European patent 2016

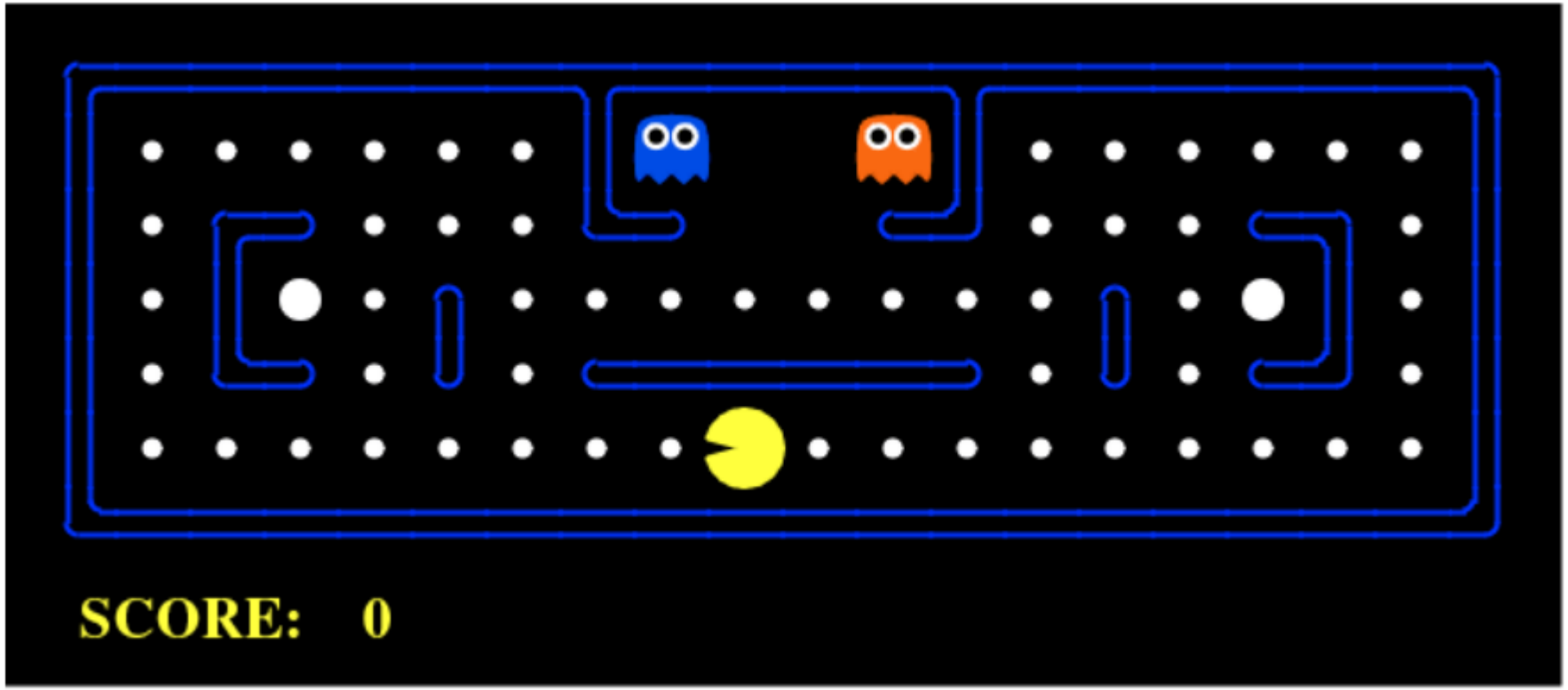
- A good summary paper:
  - Behavior Trees in Robotics and AI, Colledanchise and Ögren, arXiv preprint 2017

Main Concepts of BTs

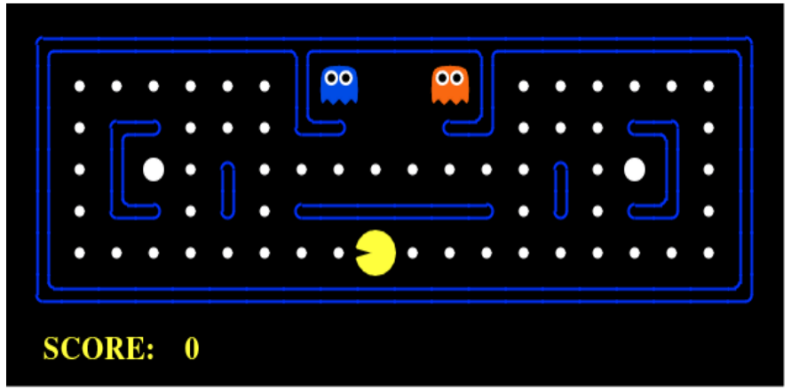
- BTs are directed rooted trees where:
  - Internal nodes (the ones with children) are called *control flow nodes*
    - *Decorator* if only one child
    - *Composite* if multiple children
  - Leaf nodes (the ones without children) are called *execution nodes*
    - *Action* if the node describes an action to be executed
    - *Condition* if the node describes a condition to be verified

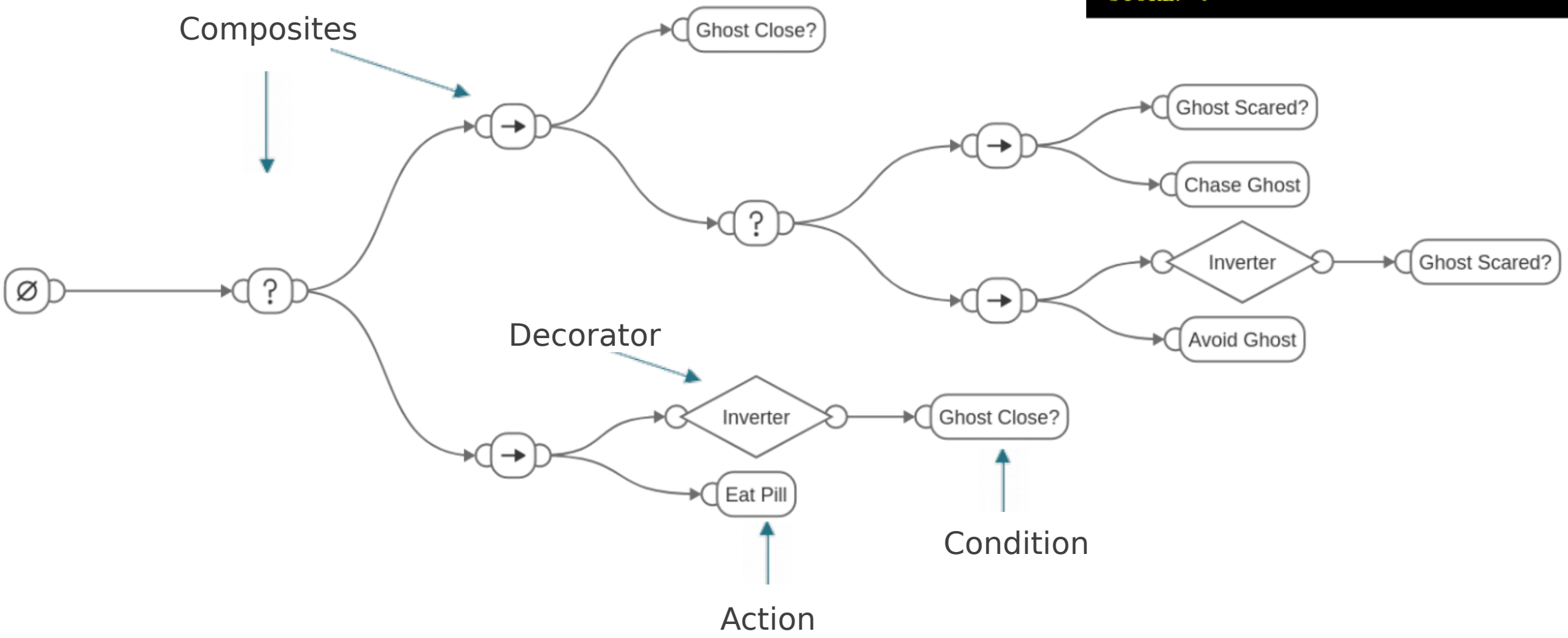
- Use the terminology of parent and children nodes
- The root node is the only one without parents

A Simple Case Scenario



A Behavior Tree for the Scenario





How is a Behavior Tree Executed?

- A BT starts its execution from the root node
- The root generates signals called ticks with a given frequency
- The ticks are propagated to the children following specific rules

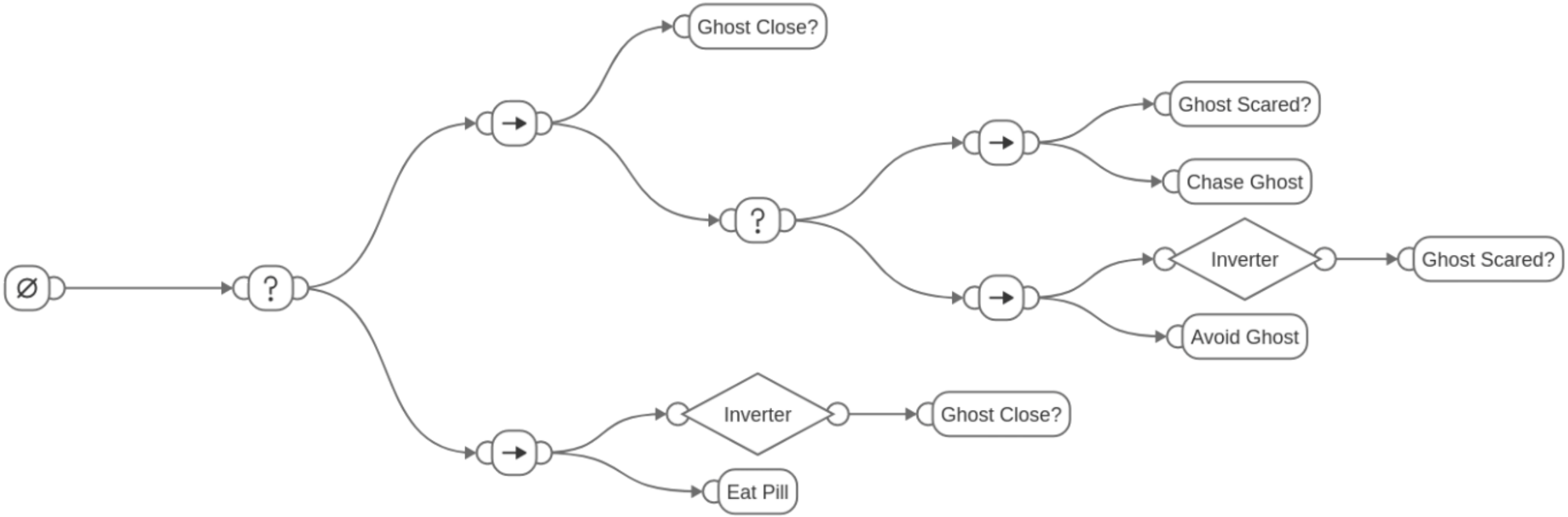
- The child returns to the parent:
  - Running, if its execution is under way
  - Success if it has achieved its goal
  - Failure otherwise

Node type	Symbol	Succeeds	Fails	Running
Fallback	?	If one child succeeds	If all children fail	If one child returns Running
Sequence	$\rightarrow$	If all children succeed	If one child fails	If one child returns Running
Parallel	$\rightrightarrows$	If $\geq M$ children succeed	If $> N - M$ children fail	else
Action	text	Upon completion	If impossible to complete	During completion
Condition	text	If true	If false	Never
Decorator	$\Diamond$	Custom	Custom	Custom

- Nodes can share information using a blackboard

Our Behavior Tree Execution

Node type	Symbol	Succeeds	Fails	Running
Fallback	?	If one child succeeds	If all children fail	If one child returns Running
Sequence	→	If all children succeed	If one child fails	If one child returns Running
Parallel	⇉	If $\geq M$ children succeed	If $> N - M$ children fail	else
Action	text	Upon completion	If impossible to complete	During completion
Condition	text	If true	If false	Never
Decorator	◊	Custom	Custom	Custom

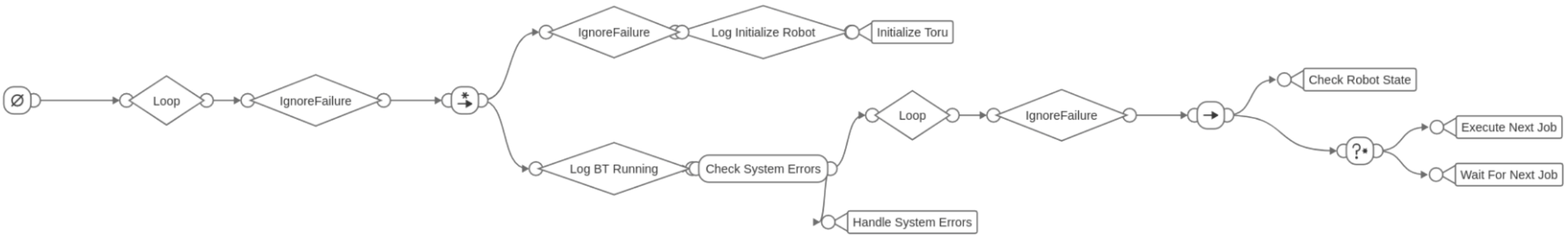


Behavior Trees at Magazino

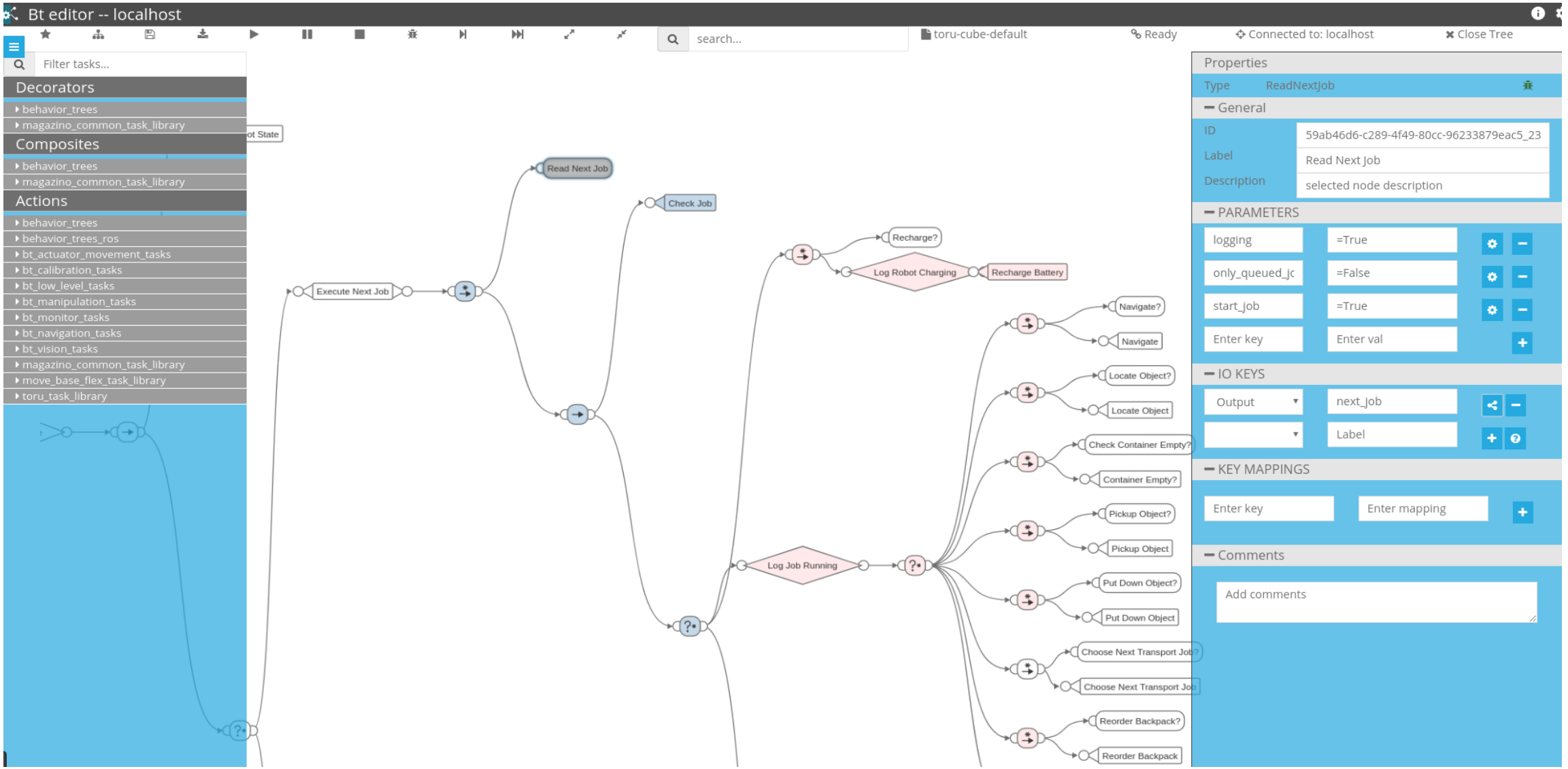
- New execution semantics
  - Memory Nodes
  - Parallel One, Parallel Selector
  - Recovery
  - Check System Errors
  - ...
- ROS integration (Often conditions as topic listeners, actions as action clients)
- Copied and scoped variables
- Subtrees
- Watchdogs
- ...

Controlling Process of Robots Having a Behavior Tree Architecture, Tenorth, European patent 2016

A More Robotic Behavior Tree



Behavior Tree Editor



An application of Behavior Trees

Why Behavior Trees?

- In comparison to Finite State Machines, BTs are much easier to adapt:
  - new branches can be integrated into a BT by adding a single connection
  - FSMs require connections for all permitted task transitions

- As Petri Nets are alternatives to FSMs emphasizing concurrency, BTs emphasize modularity

- Variables are copied and scoped, which is more manageable than having global variables

- BTs have a natural graphical representation that can be used for:
  - editing robot behavior without programming
  - visualizing the resulting behavior specification
  - inspecting the state of the control program at execution time
  - debugging behavior faults

- BTs are reactive to events: cheks at every tick instead of checking at the end of actions

- More pragmatic than planning: shortcuts can be applied more easily

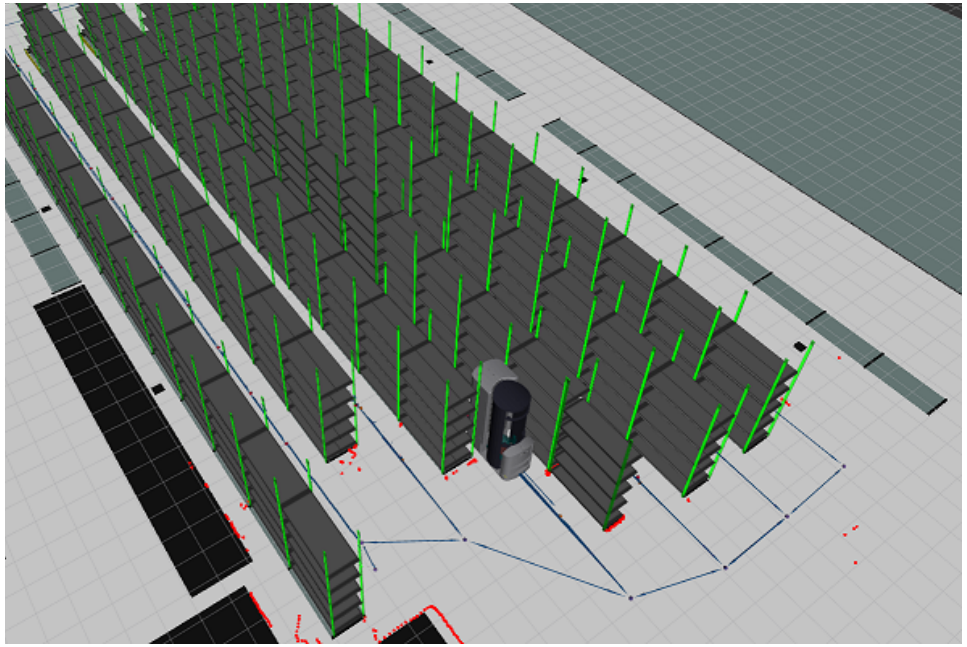
Drawbacks of Behavior Trees

- BTs operate in a recursive manner. Computationally, this could produce stack overflows
- For each tick, a large number of checks might have to be performed over the state spaces
- Different subtrees in the tree might require different frequencies
- Hard to model mutually dependent parallel actions that share information
- Less powerful but more manageable than other execution frameworks (e.g., CRAM)

Summary

Logistic environments full of:

- Enormous amount of scenarios to be faced
- Different customers to handle
- Complaints coming from customers
- Multiple type of robots



We were looking for an executor able to cope with multiple requirements

Behavior Trees gives us:

- Flexibility
- Adaptability
- Introspectability
- Reusability and scalability

We are looking for talents!

Software/Robotics

Software Architect in Python

Autonomous Navigation Engineer

Robot UI and Frontend Developer

Robot Software Enthusiast

Student Intern

... and others!

More openings on https://www.magazino.eu/jobs-2/?lang=en

Thank you for your attention

Your contact person at Magazino

Dr. Guglielmo Gemignani

Teamlead

Behaviors & Reasoning

Mail: gemignani@magazino.eu

Join the team!

For more details visit our website www.magazino.eu

Insights of Magazino

The KADO Vision System

MAGAZINO

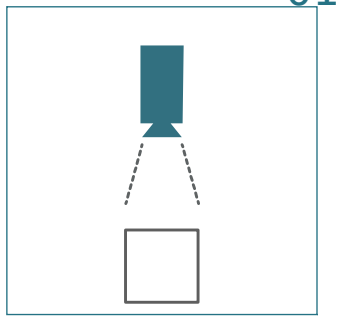
Erkannte Objekte

WildeRauke
Zitronenmelisse
Liebstöckel
Basilikum
Koriander

Advantages of Kado

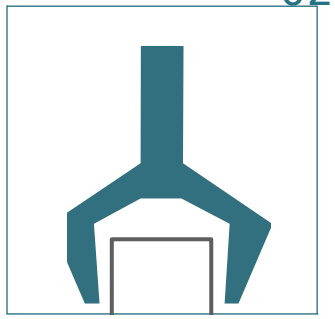
MAGAZINO

01



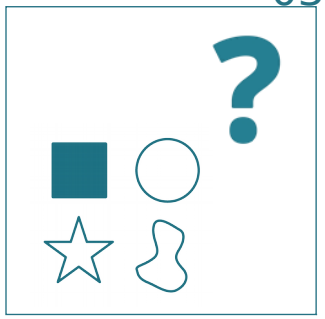
Recognize and meausure

02



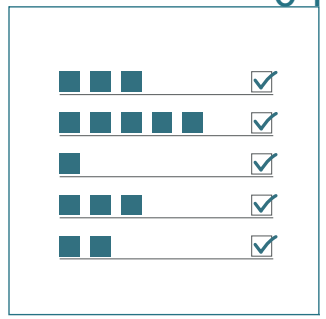
Optimized grasping points

03



Recognition without teaching-in objects

04



Easy and transparent organization

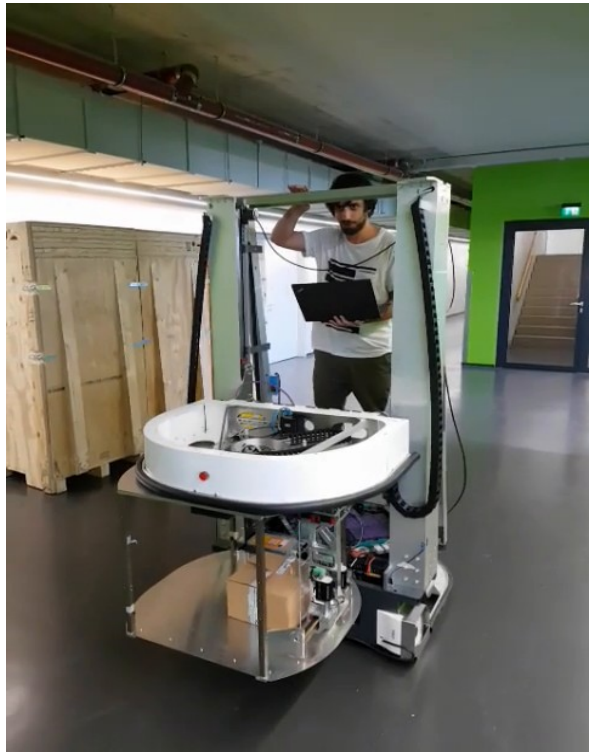
Robots move from production to testing area

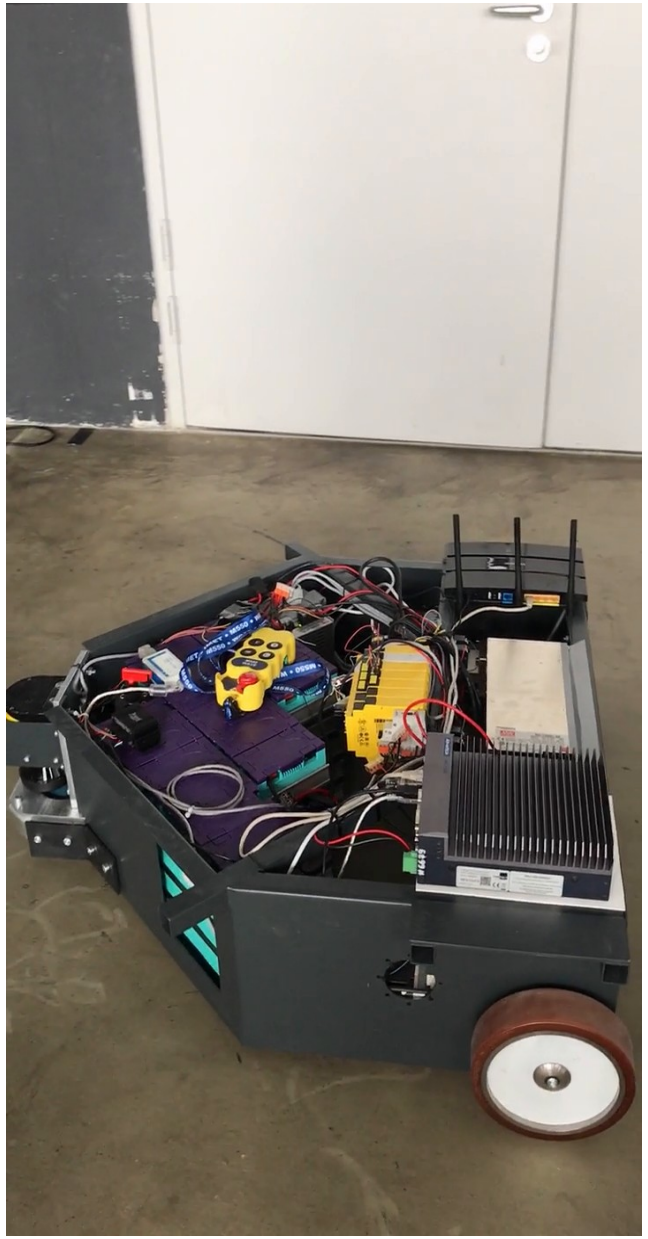
Full integration testing of hardware and software

- Reproduction of customer environment
- Automatic update of robot software over wifi
- Fake customer server sends requests to the robot
- New robots are tested using stable software
- New software is tested using a stable robot
  (confirmed to have working hardware)

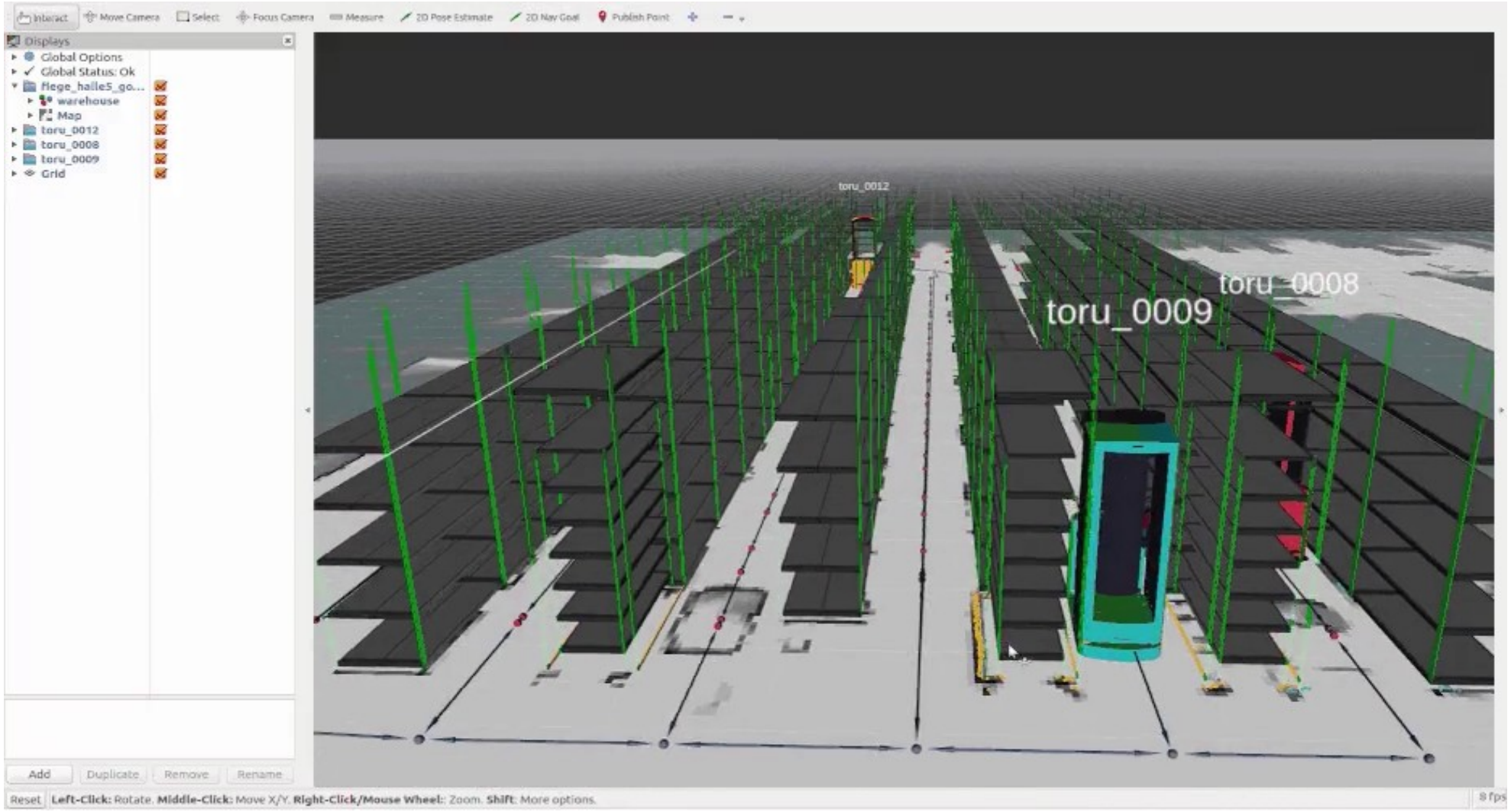
Prototyping

- Hardware robots released every 4 months
- Electrical PCB and cabling released every 2 months
- Firmware running on PCB released every month
- Software released weekly
- Customer robots receives software updates every two weeks



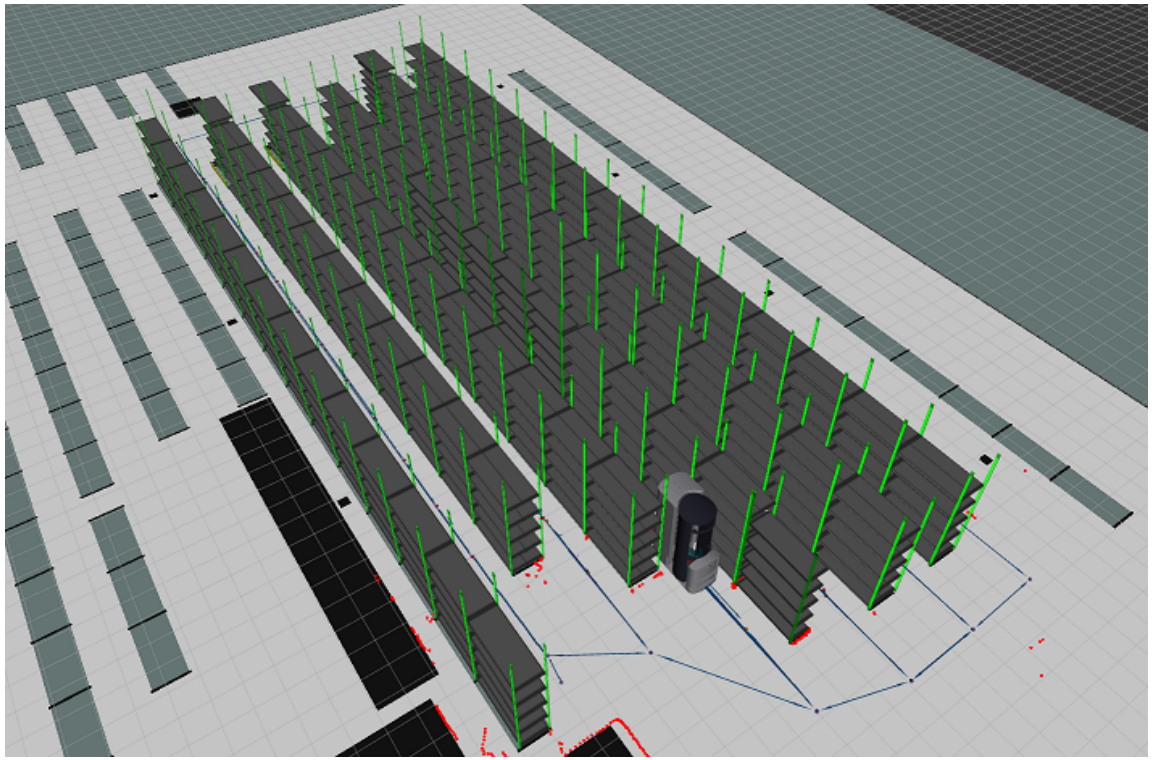


Cartographer: active collaboration between Google, Lyft, Magazino and Fetch



Remote robot visualization

- Robots live in a virtual world which is a replica of the customer warehouse
- The 3D visualization of the robot world is accessible over internet
- You can follow robots remotely! (let's try a live demo)



Customer statistics

We collect data about speed and errors from every customer