PAL

ROSCon 2024

Building Humanoid Robots: Mastering Design and Control with ROS

Luca Marchionni, CTO

ODENSE, DENMARK | OCTOBER 2024

Who we are

Associations we are part of

eu ROBOTICS

Francesco Ferro
Industry Director

5G PPP

adra
The AI Data Robotics Association

Francesco Ferro
Industry Robotics Director

Hispa Rob.

Francesco Ferro
Board Director

A3S R

Francesco Ferro
Board Director

IFR International Federation of Robotics

Francesco Ferro
Chair IFR Service Robot Group

Humanoid robots' family is growing


ASIMO


ATLAS


DIGIT


HRP 5P


PAL REEM-C


DRACO


DLR TORO


Xiaomi CyberOne


Kawasaki


Bruce


NASA VALKYRIE


PAL TALOS


IIT WALK-MAN


TUM LOLA


JSK JAXON3-P


IHMC Nadia


MIT HUMANOID


Figure 01


Expedition A1


UCLA Artemis


Apptronik Apollo


GR-2


Tesla Optimus


Unitree H1


Sanctuary AI


Kepler


PAL KANGAROO


RH5


1X Neo


NEW ATLAS


Unitree G1


UBtech Robotics' Walker S


LimX Dynamics' CL-1

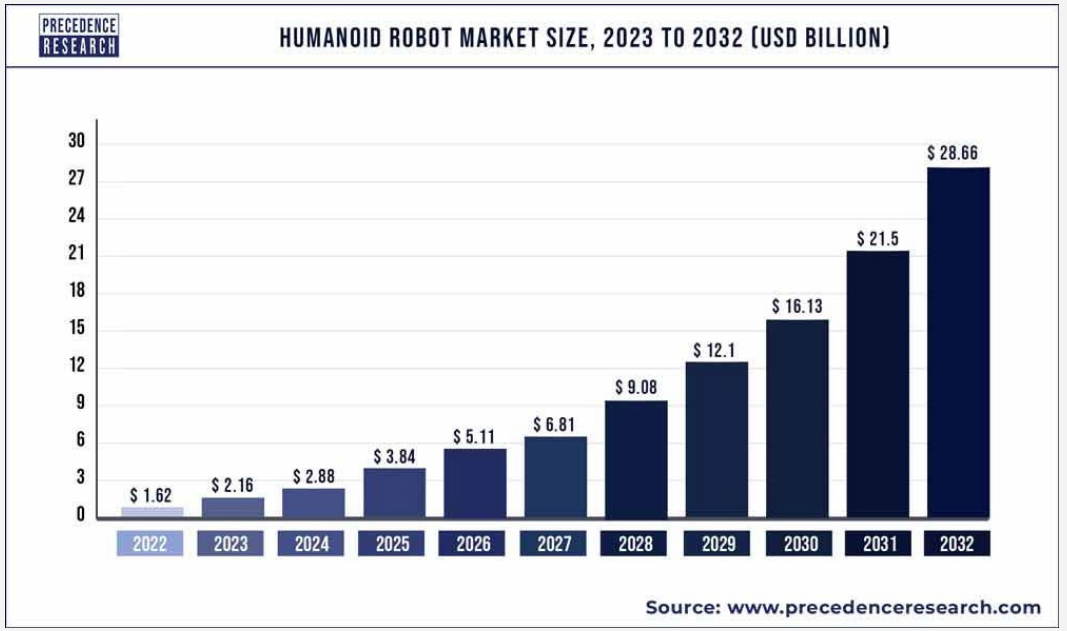

SUPCON Navigator


Tiangong


Neura 4NE-1

The rising global interest in humanoid robots

UBTECH			
1.34 B	1.1 B	?	754 M
Agility Robotics	UNITREE		
180 M	155 M	126 M	32.8 M



That's the appeal of humanoid robots, versatility over efficiency.

Since the world is built for humans a human shape machine would be very versatile.

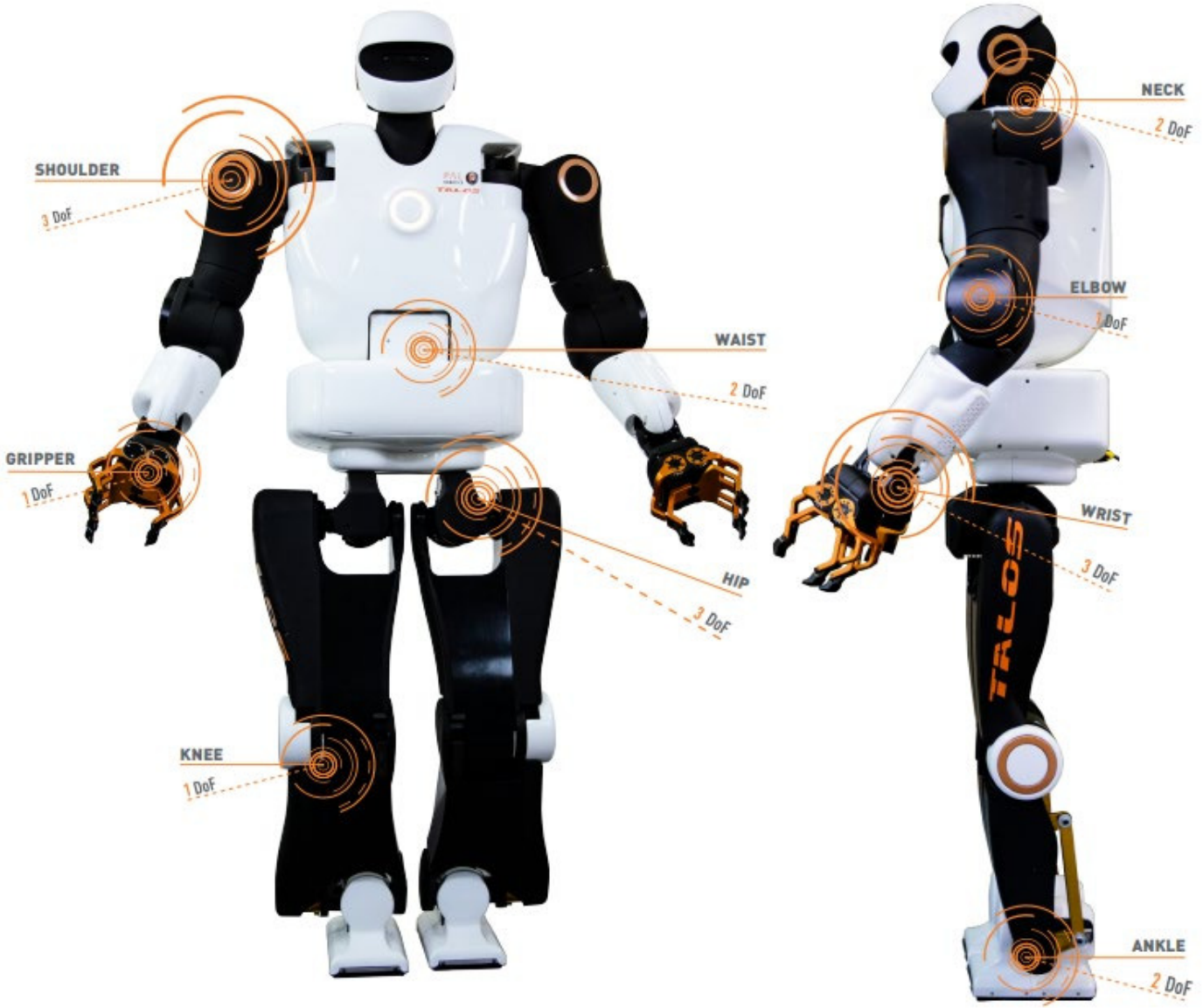
Will humanoid deployment become cheaper than retrofitting for AMRs or cobots?

Humanoids in the Market
Our Humanoids

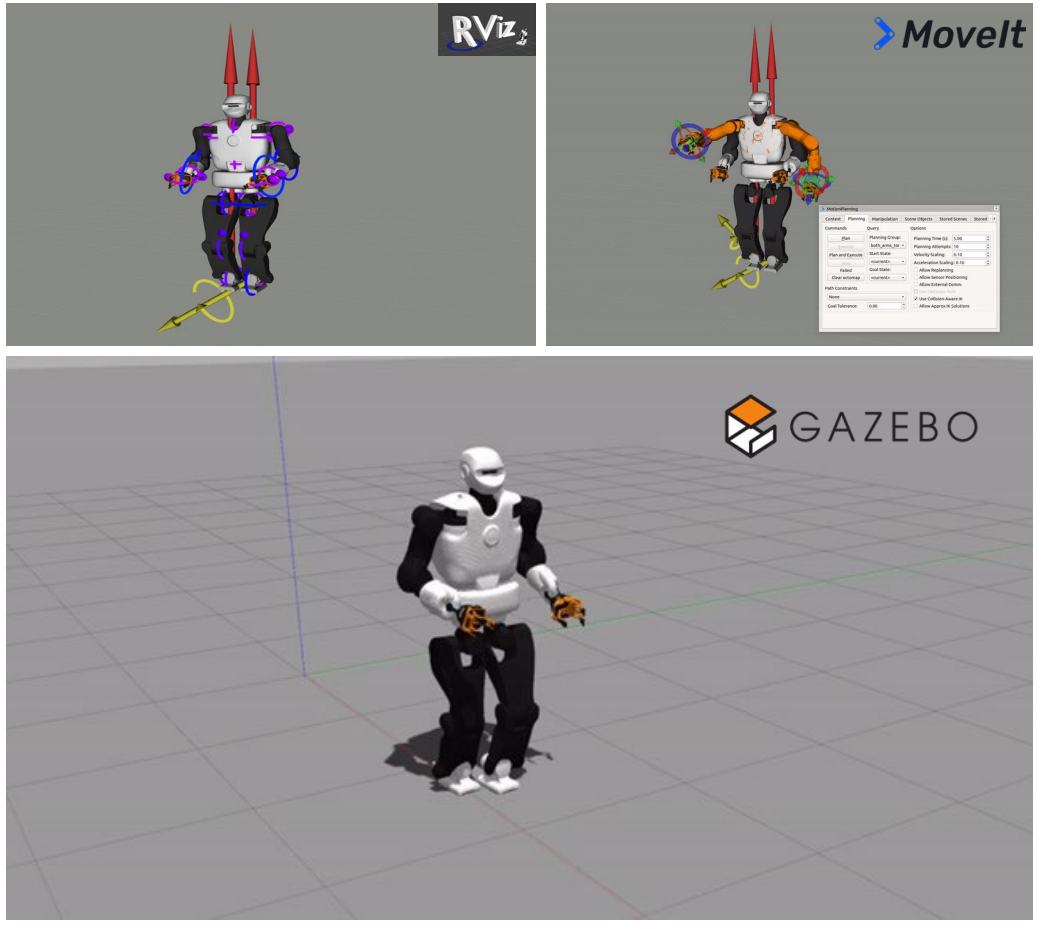
TALOS

High Payload Biped

100% ROS	**Torque** Controlled	**6 axis** FT Sensors ankle/ wrists
6 Kg Payload per arm	**32** Actuated DoF	**EtherCAT** control loop 2kHz
175 cm Height	**95 kg** Weight	**3 hours** Autonomy



TALOS with ROS



Robot Model and Sensors Visualization
Motion planning and grasping
3D Physics Simulation

TALOS Robot in action

TALOS demonstrations

MPC walking with WBID

Grasping and crossing debris

Introduction video

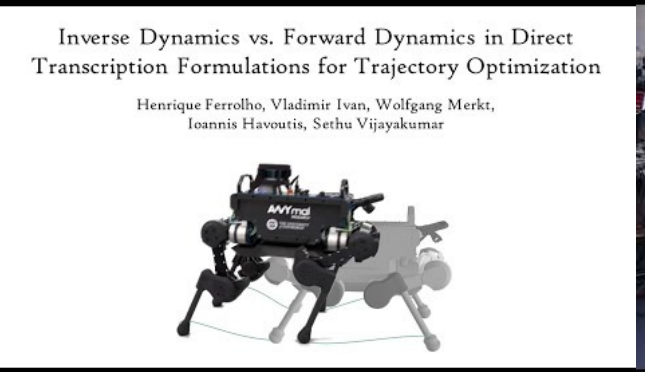
Torque controlled
Centroidal MPC walking

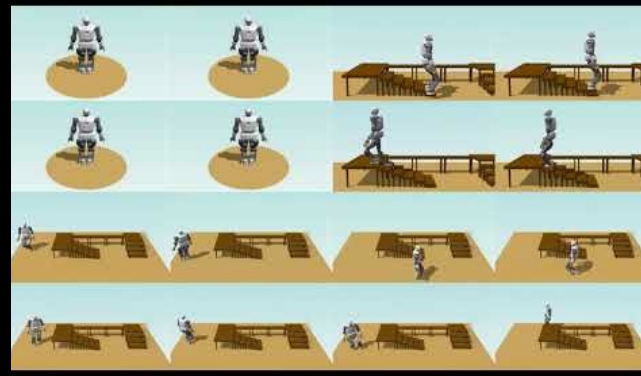
TALOS grasping
and walking

TALOS fast swing
leg motion

Torque controlled
Whole Body Balancing

Research with TALOS. Our Customers



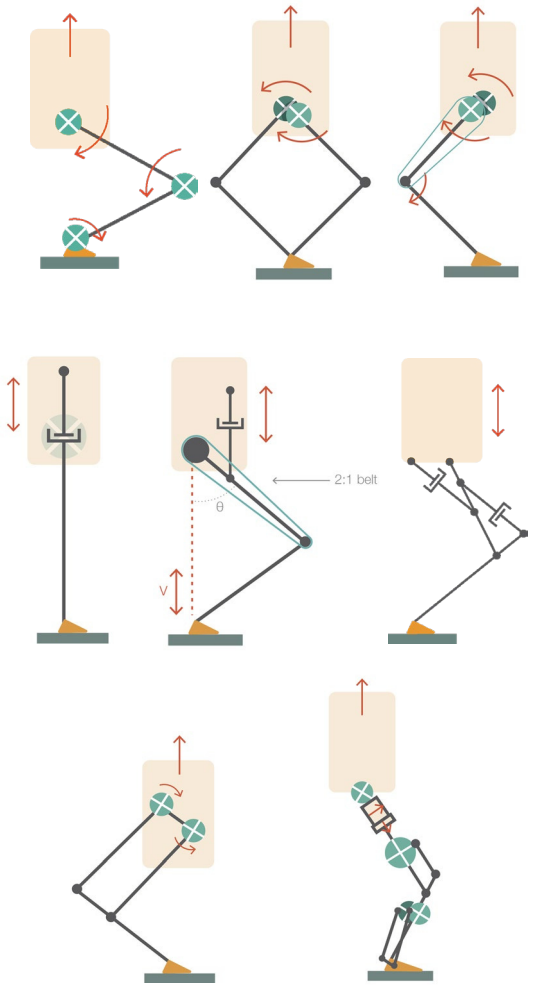


KANGAROO

Agile, Dynamic and Robust Biped

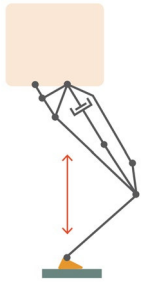
100% ROS	**40 kg** Weight	**Torque** Controlled
64 Passive DoF	**12** Actuated DoF	**EtherCAT** Control loop 2kHz
160 cm Height	**Force sensors** each actuator	**3 hours** Autonomy

In the market
Leg architectures

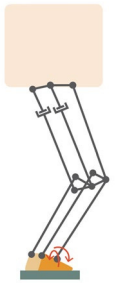


KANGAROO'S
NEW architecture

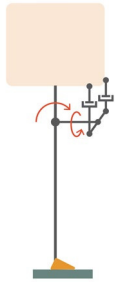
Leg length



Ankle



Hip



Leg mass optimization and low inertia

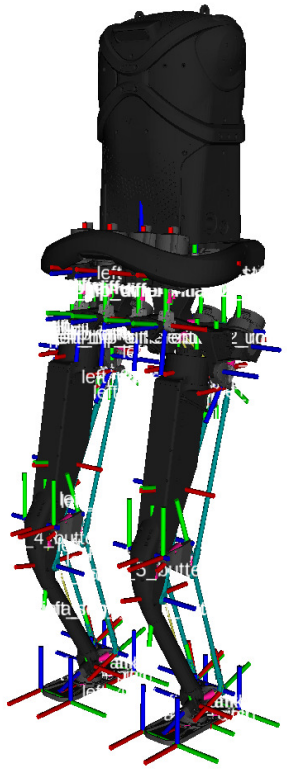
Reduce energy cost for walking
[Browning et al., 2007]

Shorten swing time
[Royer and Martin, 2005]

Reduce non-linear effects of swing leg

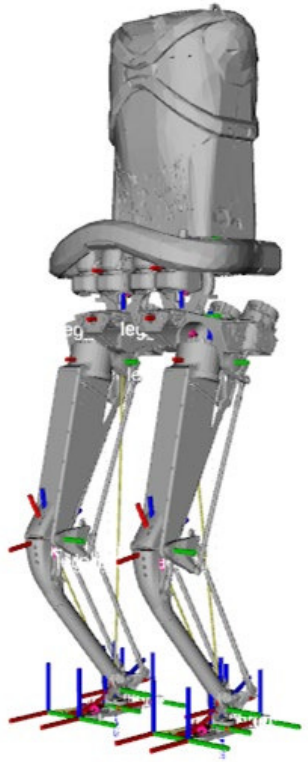
KANGAROO with ROS

Full model



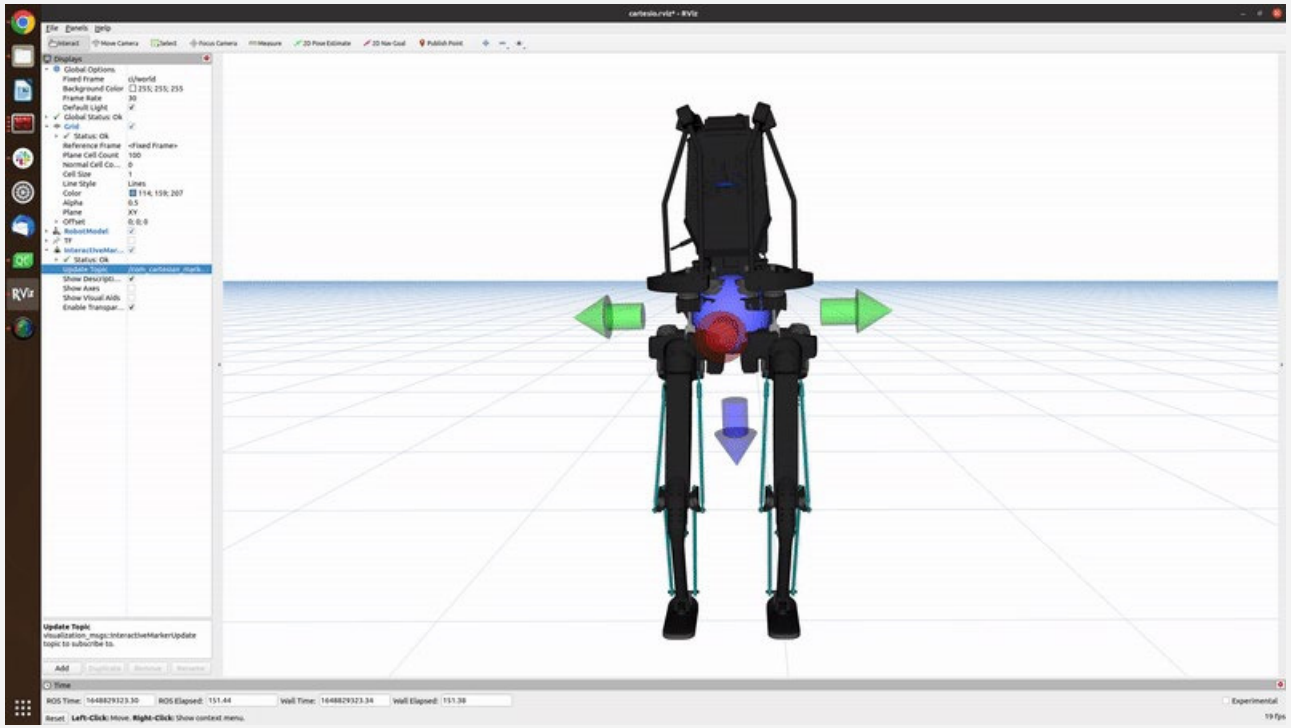
12 active DoF
64 passive DoF

Simplified URDF



12 active DoF
4 passive DoF

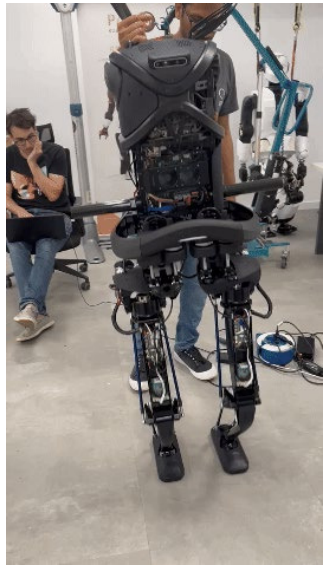
KANGAROO
operational space control from Rviz



Torque controlled balancing

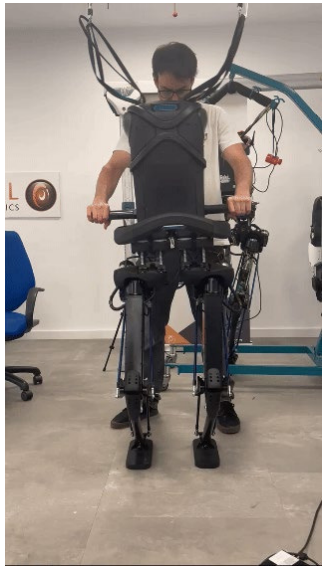
Free walking

Position control walking and stabilizer

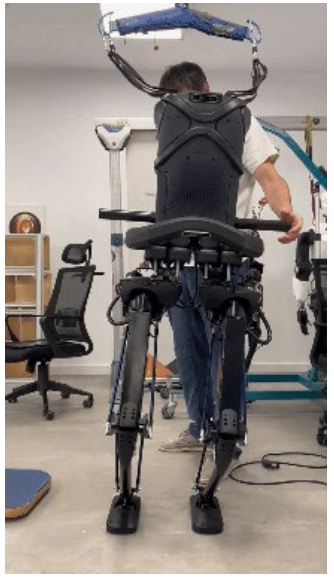


MPC walking

Fast squat



Compliant control



Balance shifting

One leg stand

MPC walking

KANGAROO Control Stack *on going development*

Optimal control

Robot Trajectory Optimisation (RTO)

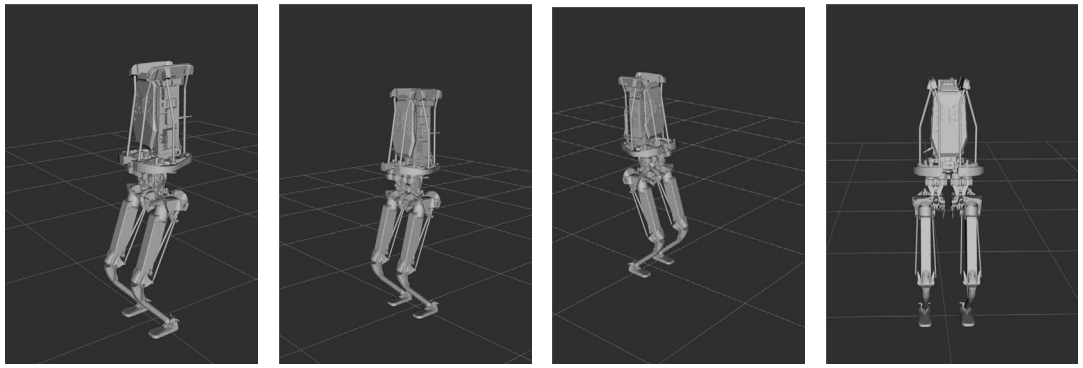
↓

Model Predictive Control (MPC)

↓

Whole Body Inverse Dynamics (WBID)

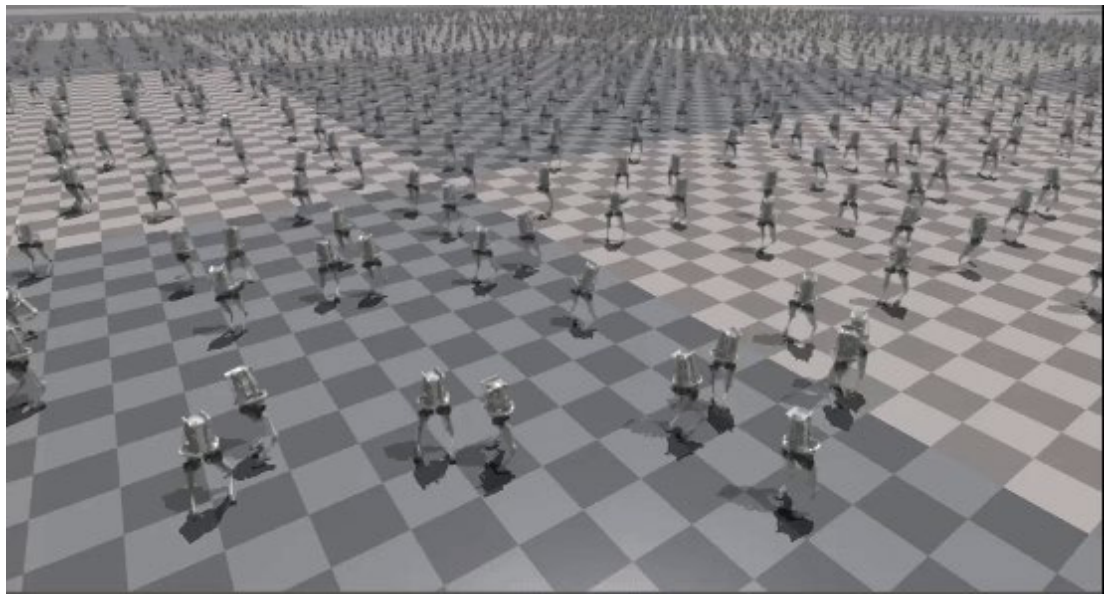
↓



Reinforcement Learning

RL framework

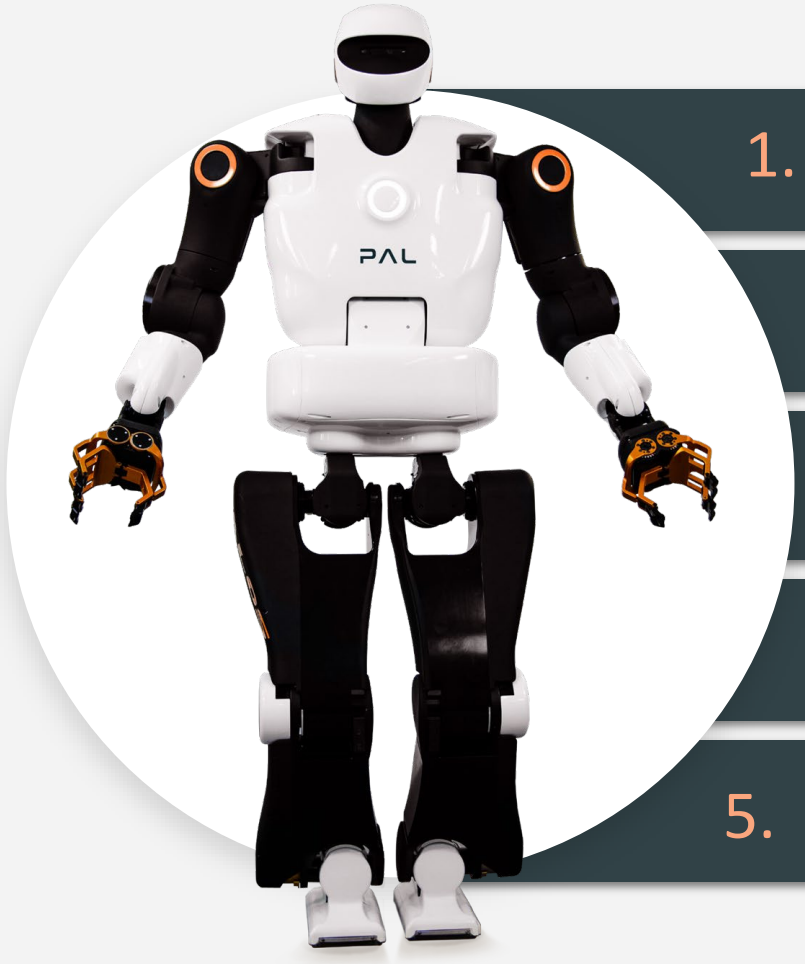
4000+ models in parallel, hardware-accelerated, trained policies in ~20 min, pipeline using **Isaac Sim, MuJoCo.**



What do we need
to control a humanoid robot?

We need real-time

Why?



1. DETERMINISM
2. BALANCE AND STABILITY
3. MOVEMENTS' COORDINATION
4. DYNAMIC ENVIRONMENT
5. SAFETY

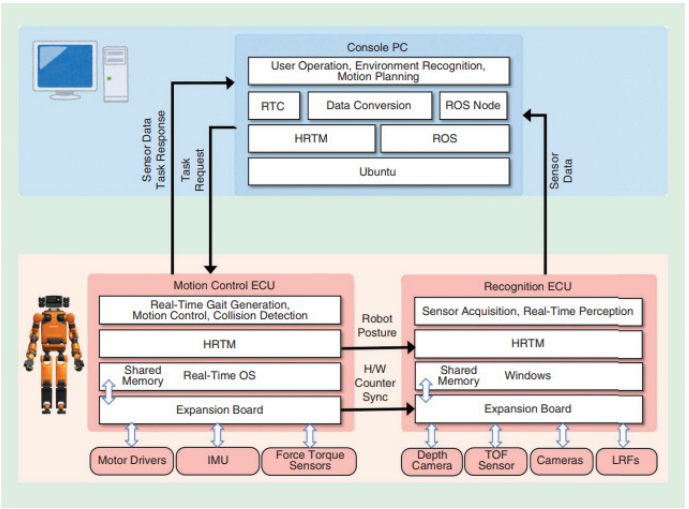
©PAL Robotics 2024. All rights reserved

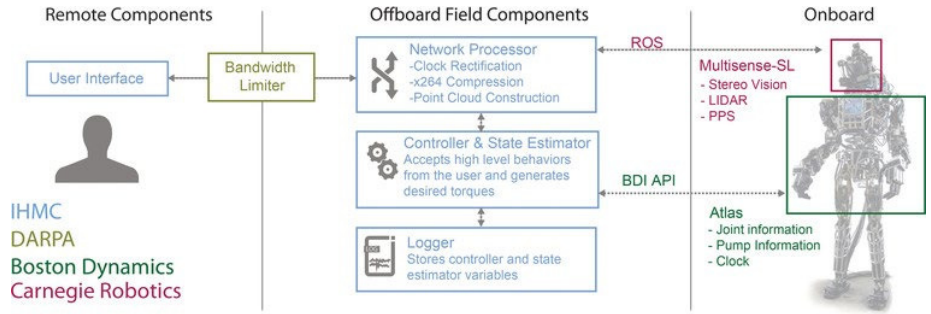
How we make it work?

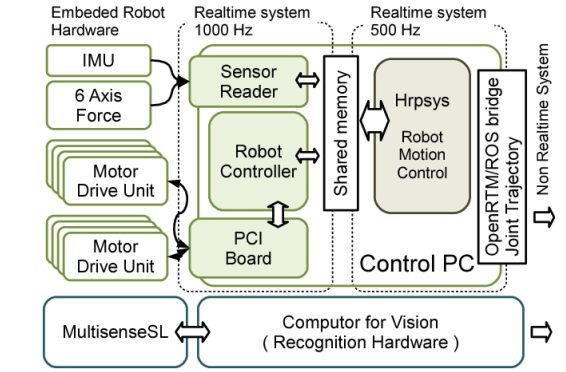
Architecture Layers

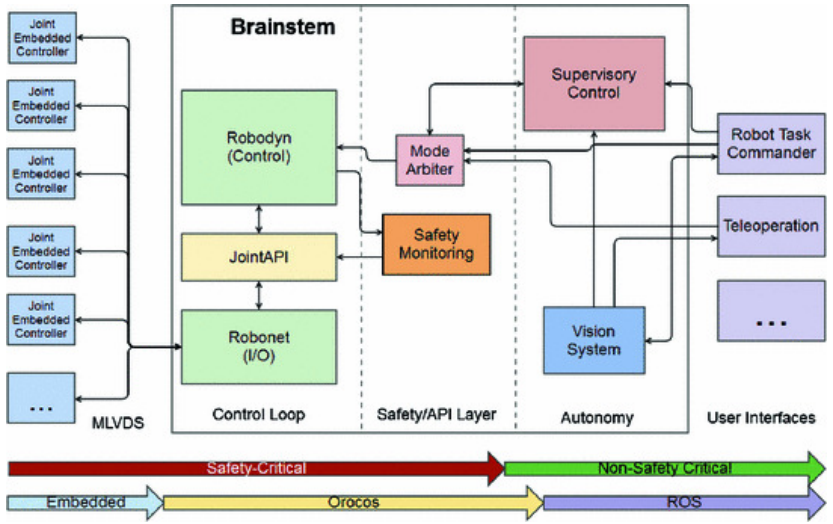
→ **High level app**	Perception, Motion Planning, FSM, BehaviorTree, Nav2, MoveIt2, …
→ **RT controllers**	State estimator, WBC, walking, grasping, …
→ **RT framework & RT safe comm**	Orocos, ros_control, YARP, OpenRTM, ros2_control, …
→ **Real-time System**	Hard real-time or soft real-time OS
→ **Control PC**	High-performance CPU, fast memory, reliable I/O
→ **Communication Bus**	Fast and reliable communication (EtherCAT, CAN, RoboNet…)
→ **HW devices**	Motors, sensors, …

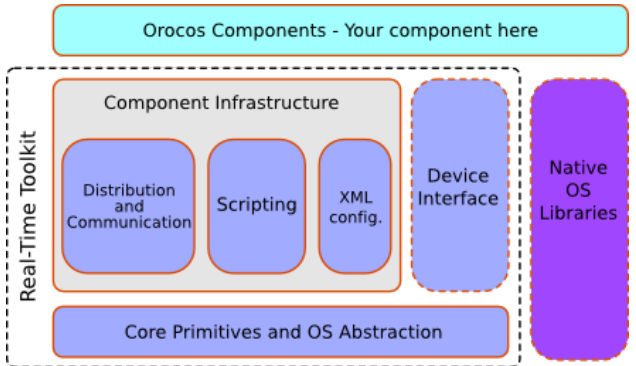
Many frameworks and architectures…

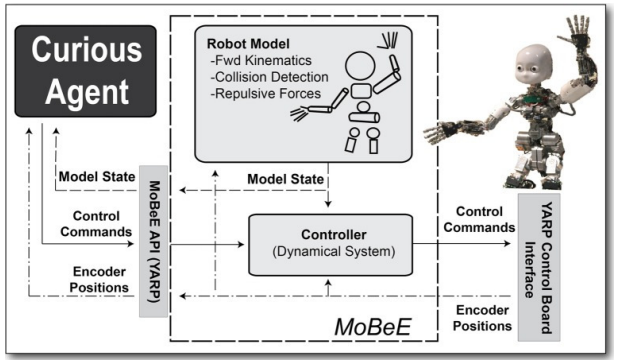


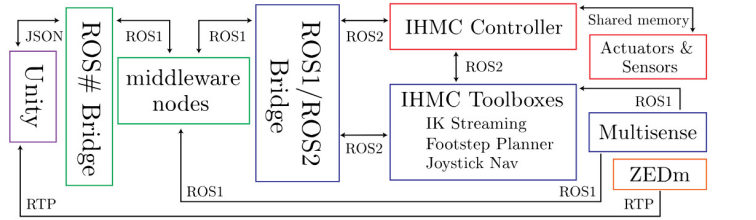


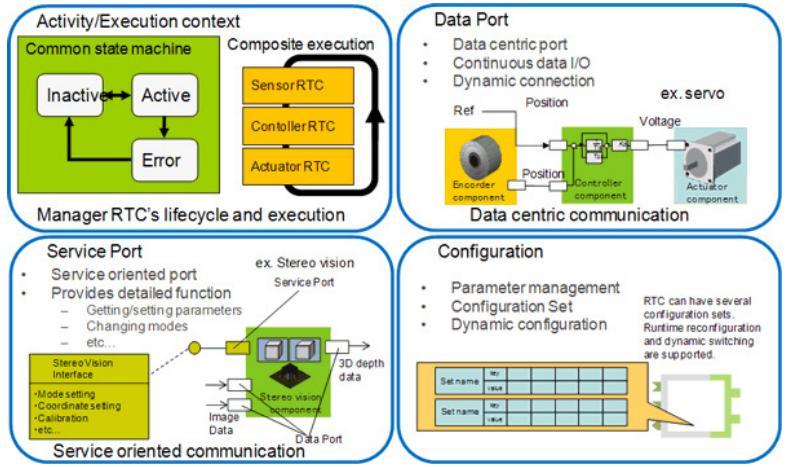








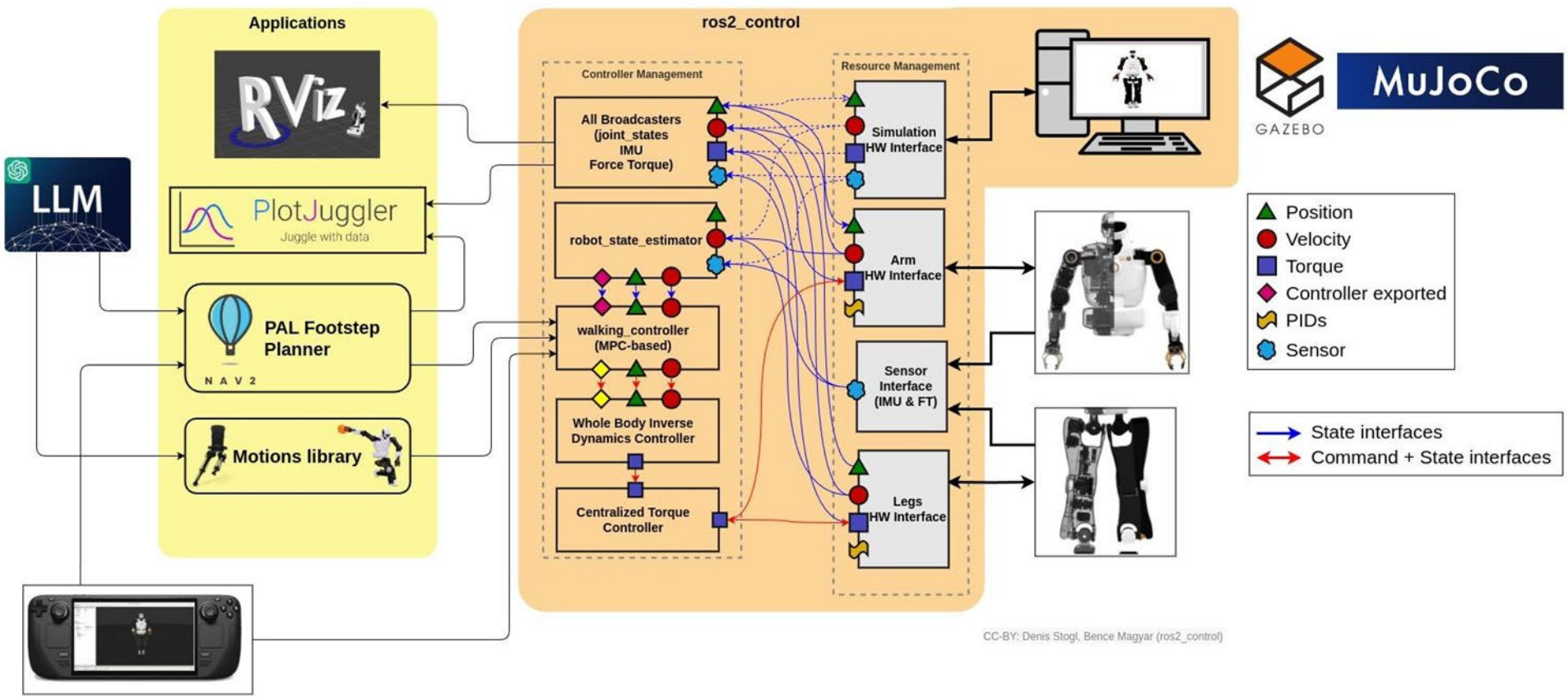




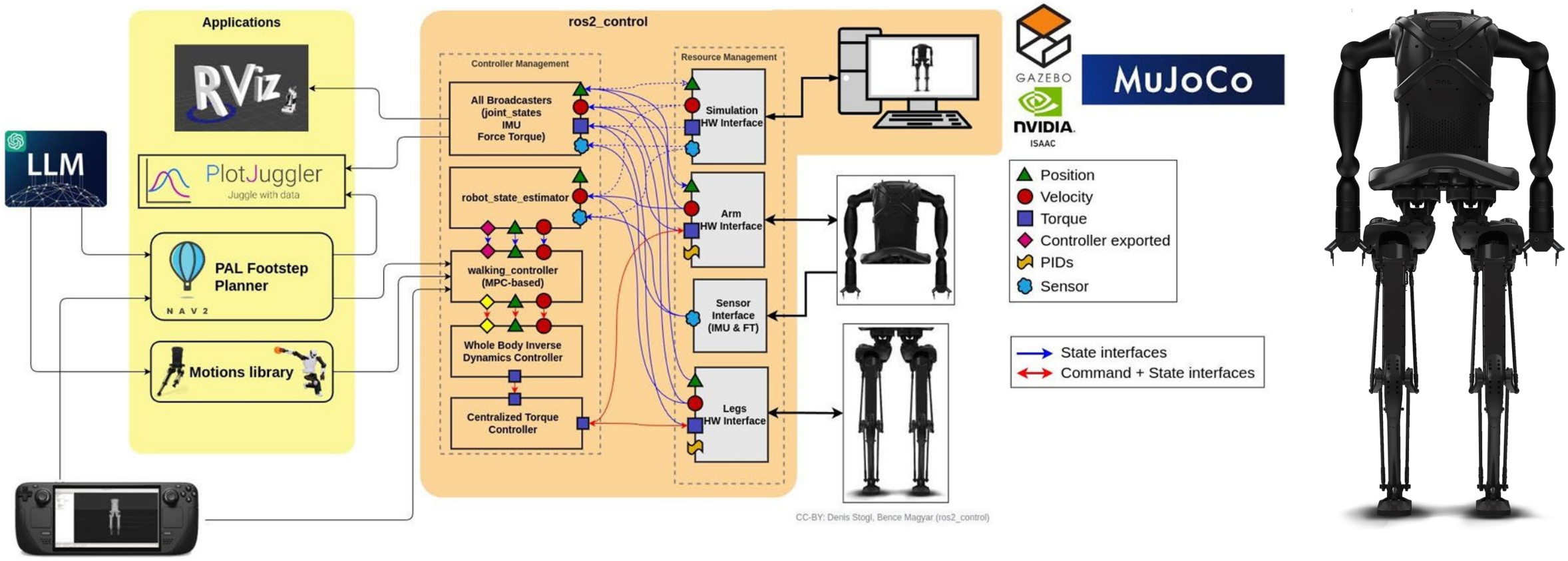
Our system architecture over the years

	Before ROS		ROS 1					ROS 2	
	2005	2008	2012	2015	2016	2018	2023	2024	2024
Architecture Layers	**REEM-A**	**REEM-B**	**REEM-C**	**REEM-C**	**TALOS**	**TALOS**	**KANGAROO**	**TALOS**	**KANGAROO**
High level app	Custom apps		ROS 1					ROS 2	
Control loop	Real-time loop @ **100Hz**		Real-time loop @ **200Hz**		RT loop @ **1kHz**	Real-time loop @ **2kHz**		Real-time loop @ **2kHz**	
RT framework & RT safe comm	Custom HAL		ros_control					ros2_control	
	Shared memory IPC	Orocos RTT	Orocos RTT						
RT OS	Elinos Linux		Xenomai Linux	Linux RT Preempt				Linux RT Preempt	
Control PC	PC 104 + Pentium M 1GB RAM		PCIe/104 SBC 4th gen i7 8GB RAM		Com Express i7 6th gen 16GB RAM		i7 9th gen 32GB RAM	i7 11th gen 32 GB RAM	i7 13th gen 32 GB RAM
Comm bus	CAN bus		CAN bus		EtherCAT			EtherCAT	
HW devices	BLDC Motors, FT Sensors, IMU				BLDC Motors, Joint torque sensors, FT Sensors, IMU				

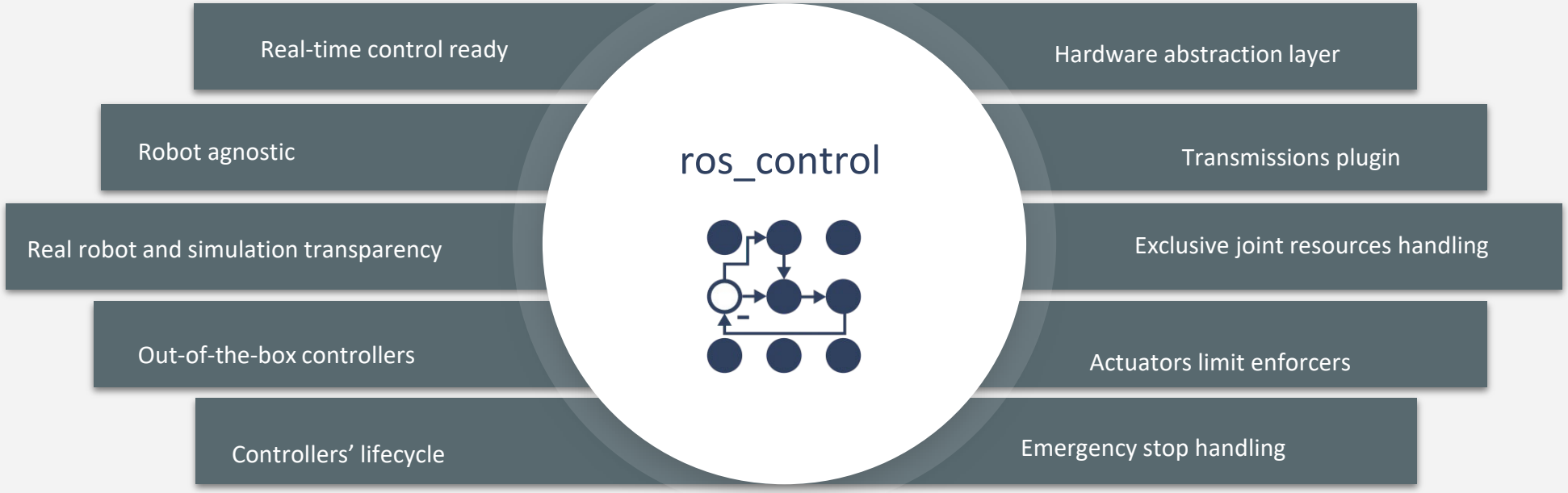
General Overview of PAL humanoids control architecture



General Overview of PAL humanoids control architecture

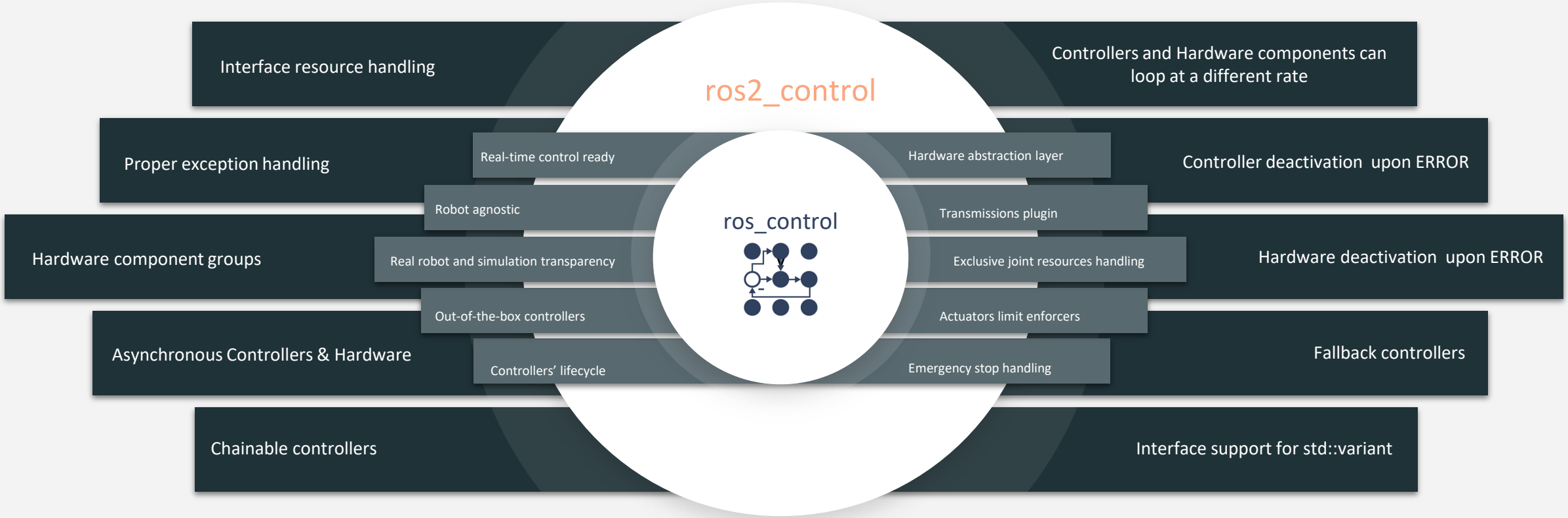


How ros_control is improving
as framework for humanoids control



©PAL Robotics 2024. All rights reserved

How ros_control is improving
as framework for humanoids control



©PAL Robotics 2024. All rights reserved

Challenges

encountered during migration to ROS 2

ros_control/ros2_control
being real-time ready doesn't mean your system will be real-time

Allowed system jitter
depending on CPU, kernel drivers and communication technology

Lack of standard introspection tools and diagnostics

Modeling, simulation and control of closed kinematic chains

Issues with the DDS/RMW configuration and tuning

ROS 2 parameters
handling in nodes and controllers (dynamic load of controllers)

Running controllers and hardware components
at different frequencies

ROS 1 Noetic → ROS 2 Humble
no features parity made us do some workarounds, leading to technical debt

Network infrastructure to handle multiple robots
connected on same network (limited ROS_DOMAIN_IDs, discovery)

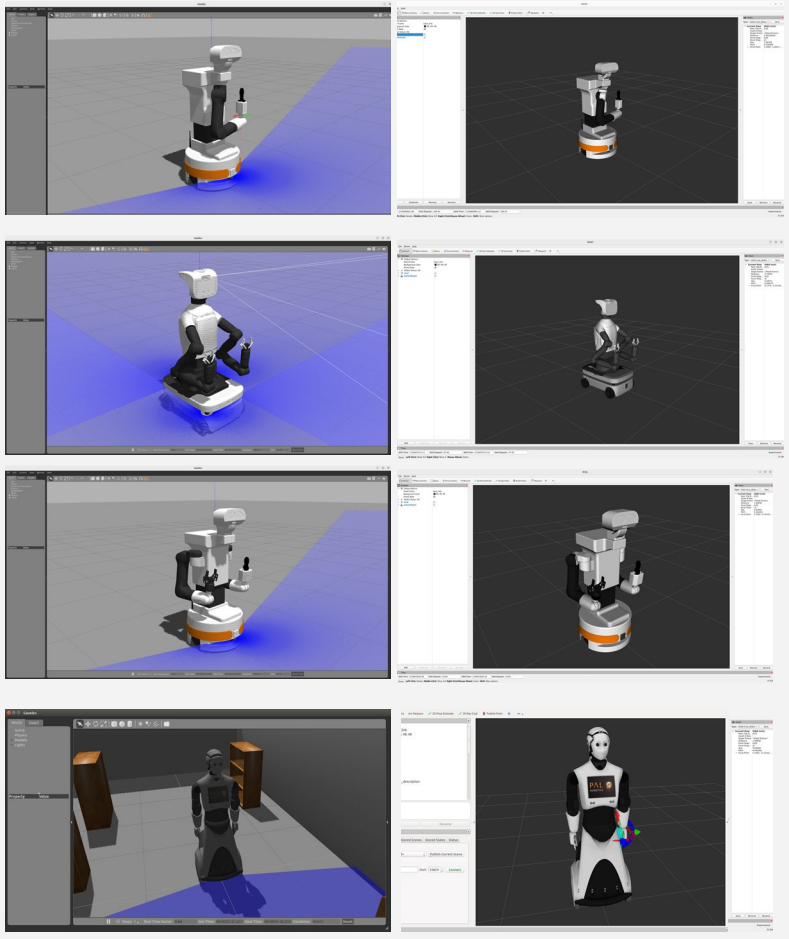
ROS 2 launch file
Flexibility comes with higher complexity and new bugs

Future

Directions with ROS 2 and beyond…

- **Redesign our libraries and algorithms to be ROS-independent**
- **Less forking and more contributing**
- **Release and maintain our robot simulations to upstream LTS distros**
- **Integrate more introspection tooling**
- **Migration to Jazzy**
- **Use Low Latency C++ Logging Library**
- **Add support for more variants of sensors, grippers and hardware**
- **Add support to multiple physics simulators**

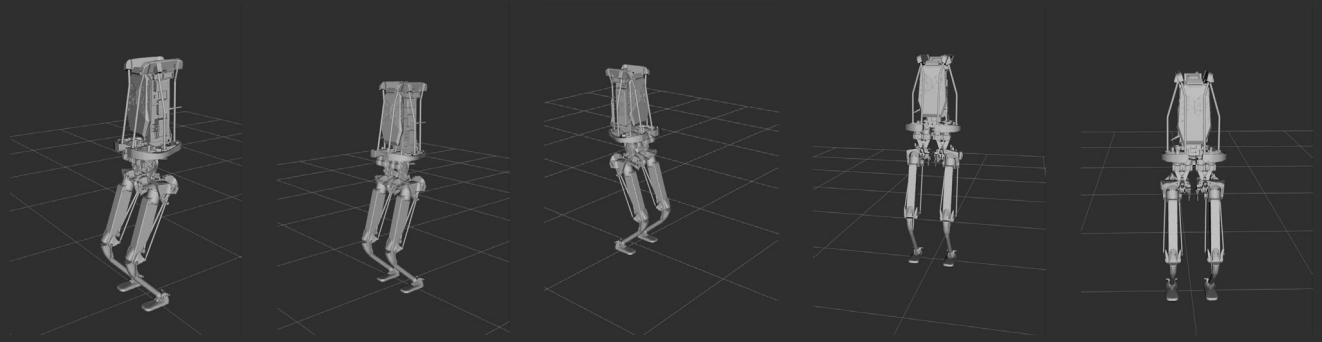
We are more than...



...humanoids

Thank you PALs, thank you ROS!

20 YEARS OF ROBOTICS



Thank you

Luca Marchionni

luca.marchionni@pal-robotics.com

PAL | 20 YEARS OF ROBOTICS

Let's build tomorrow together

Send us your CV at **recruit@pal-robotics.com**

Check our job openings

join our team