

SONY

Isaac×ROS aibo シミュレーション環境

舘石藍^{*1} 楚輝^{*2} 三本松利尚^{*1} 新原英樹^{*2}



*1 ソニーグループ株式会社 事業開発プラットフォーム 事業開発部門 AIロボティクス設計部

*2 ソニーセミコンダクタソリューションズ株式会社 設計基盤技術部門 PF技術2部

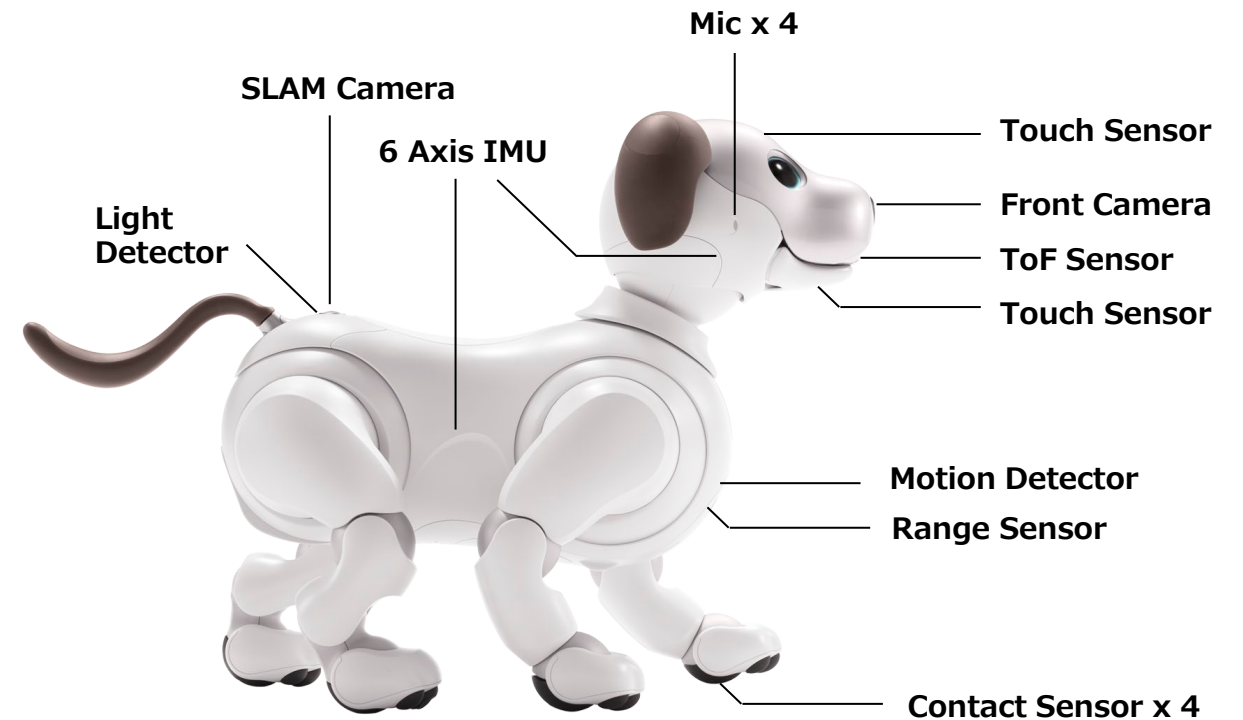
© 2025 Sony Group Cooperation

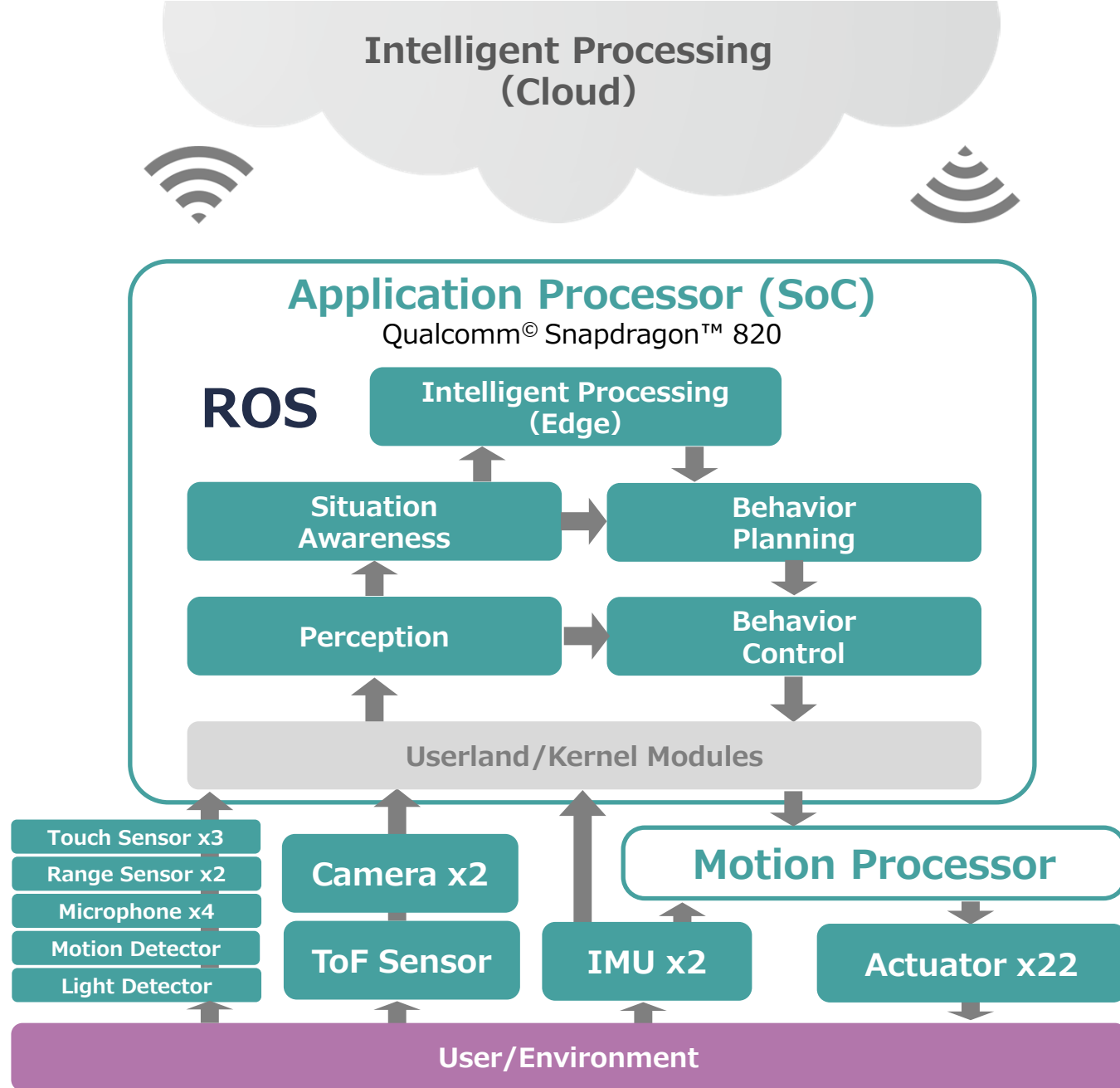
aibo



✓ 22 DoF
Legs (12), Neck, Head (3), Mouth, Ear (2), Waist, Tail (2)

✓ Many Sensors, Cameras and Microphones

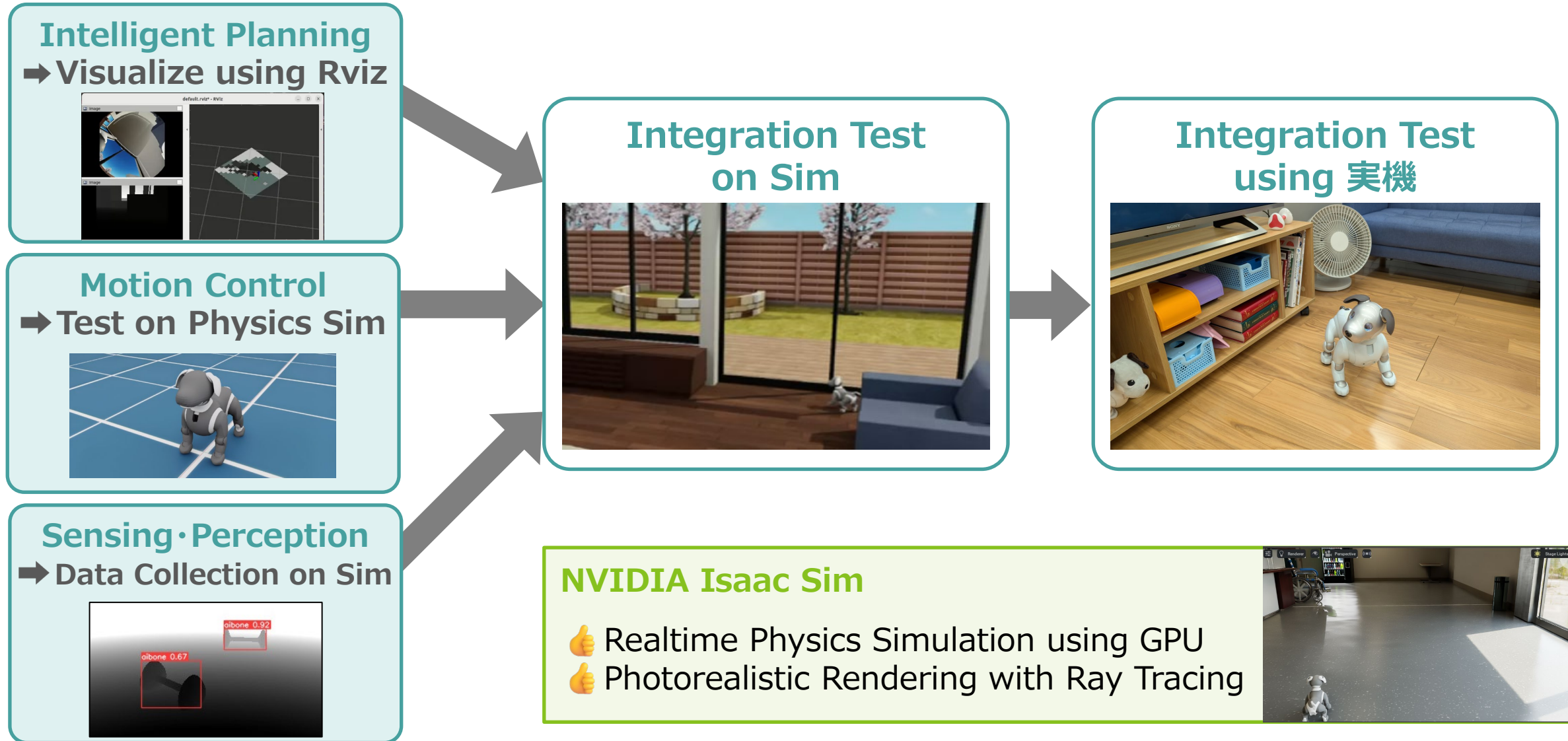




aibo



Development Workflow with Sim



Application Processor (SoC)

The diagram illustrates the architecture of the ROS-based intelligent processing system. It is organized into several layers:

- ROS Layer:** Contains the main processing components:
 - Intelligent Processing (Edge):** The top-level processing unit.
 - Situation Awareness** and **Behavior Planning:** These two components are connected by a bidirectional arrow.
 - Perception** and **Behavior Control:** These two components are connected by a bidirectional arrow.
 - Userland / Kernel Modules:** A central module that interfaces with the hardware and the ROS layer.
- Sensors and Actuators Layer:**
 - Camera x2** and **ToF Sensor:** These sensors provide input to the **Userland / Kernel Modules**.
 - IMU x2:** This sensor provides input to the **Userland / Kernel Modules** and also receives input from the **Motion Processor**.
 - Actuator x22:** This actuator receives output from the **Motion Processor** and provides output to the **User/Environment**.
- User/Environment Layer:** The base layer representing the physical world.

The flow of data is as follows: The **User/Environment** provides input to the **Sensors** (Camera, ToF, IMU). The **Sensors** provide input to the **Userland / Kernel Modules**. The **Userland / Kernel Modules** provide input to the **ROS** layer. The **ROS** layer processes the data and provides output to the **Motion Processor**. The **Motion Processor** provides output to the **Actuator**, which then interacts with the **User/Environment**.

ROS Container

The diagram illustrates the ROS-based system architecture, organized into two main functional blocks: a top-level ROS layer and a bottom-level hardware layer, separated by a horizontal line.

ROS Layer (Top):

- Intelligent Processing (Edge):** The highest-level processing block, which interacts with the ROS layer.
- Situation Awareness** and **Behavior Planning** are connected by a bidirectional arrow.
- Perception** and **Behavior Control** are connected by a bidirectional arrow.
- Motion Processor Emulator** is positioned below Behavior Planning and Behavior Control, with arrows pointing from both to it.

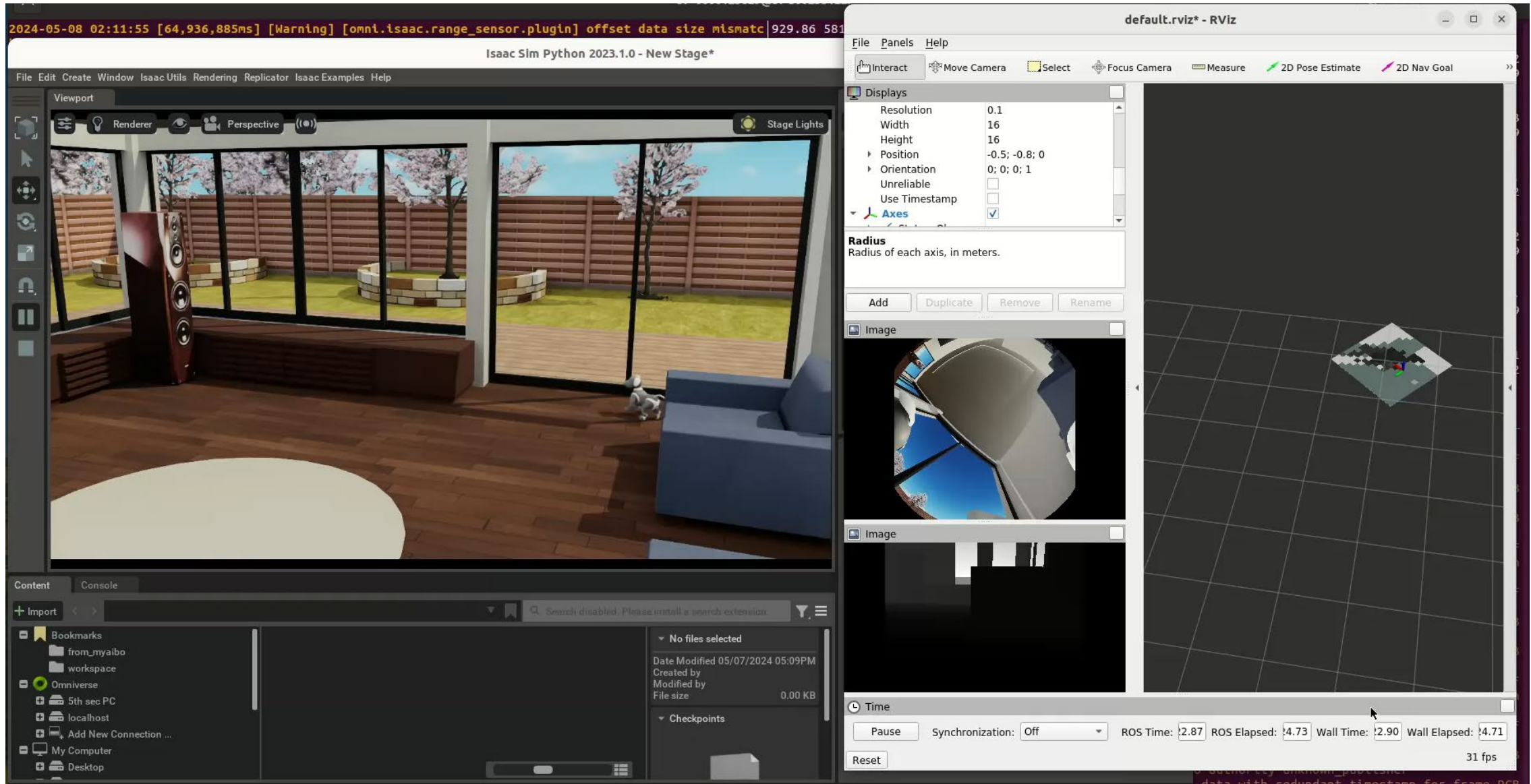
Hardware Layer (Bottom):

- Camera x2** and **ToF Sensor** provide input to **Perception**.
- IMU x2** provides input to **Behavior Planning**.
- Actuator x22** receives output from **Behavior Control**.

Human/Environment: The base layer of the system, interacting with the hardware components.

Isaac SIM Container

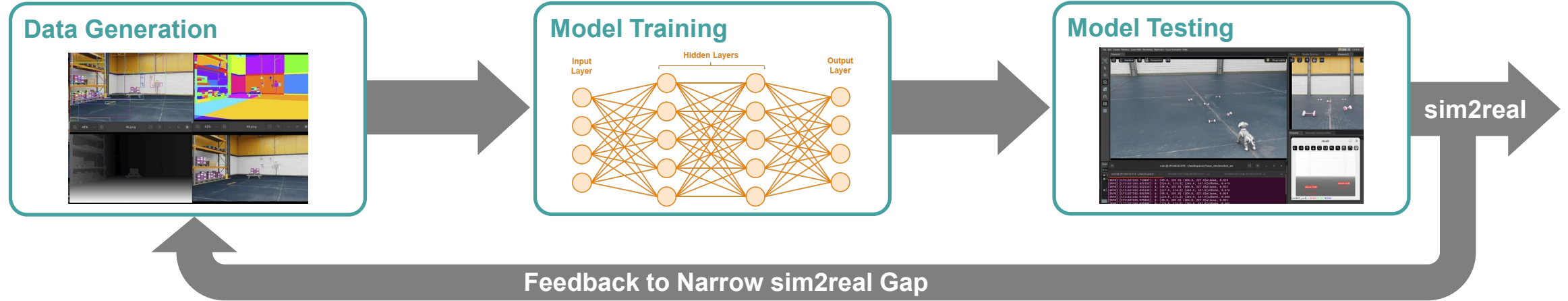
aibo Simulator



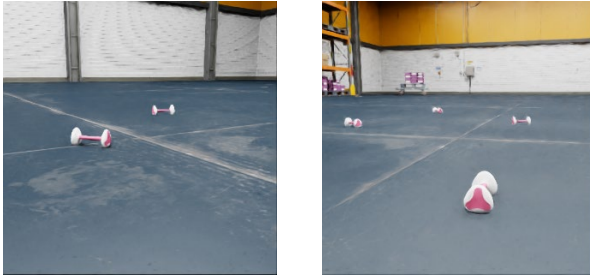
Recognizer Training with Sim



Recognizer Training with Sim



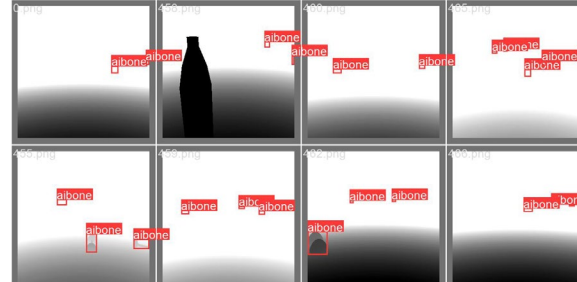
Data without Distractors



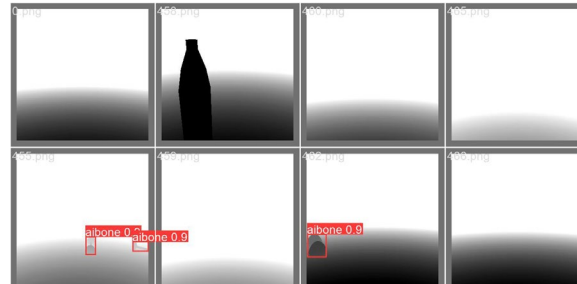
Data with Distractors



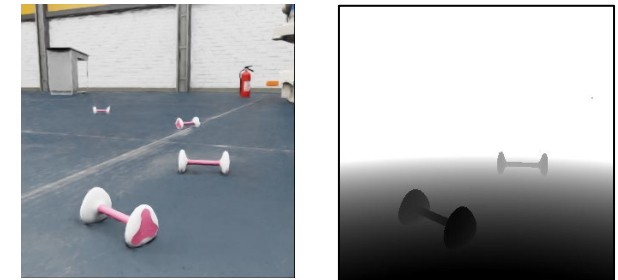
Valid Data Labels



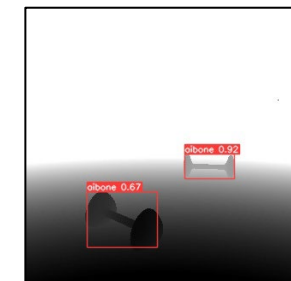
Valid Data Predictions



Realtime ROS Image from Isaac Sim

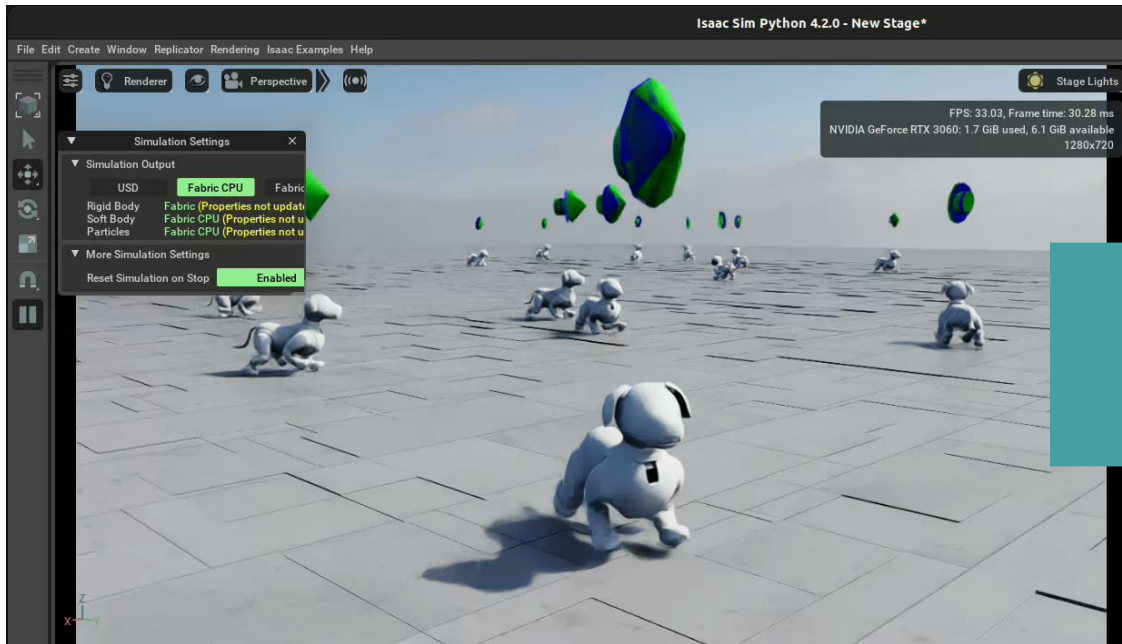


Realtime Inference Result



Motion Reinforcement Learning

Reinforcement Learning on Isaac Lab



4096 aibo robots are learning to walk in Isaac Lab.

Deploy
to
実機

Inference on 実機

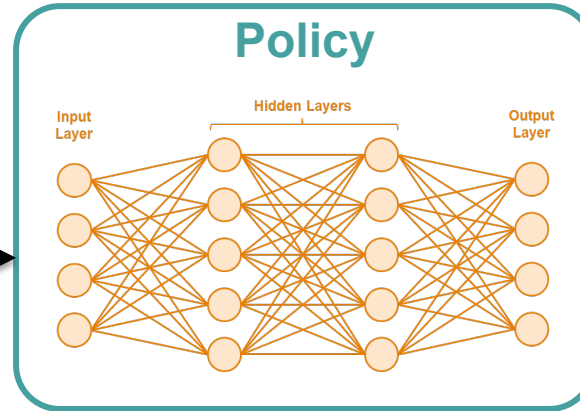


Policies trained in Isaac Lab can be directly deployed on the real robot, without any additional tuning.

Motion Reinforcement Learning

Observation

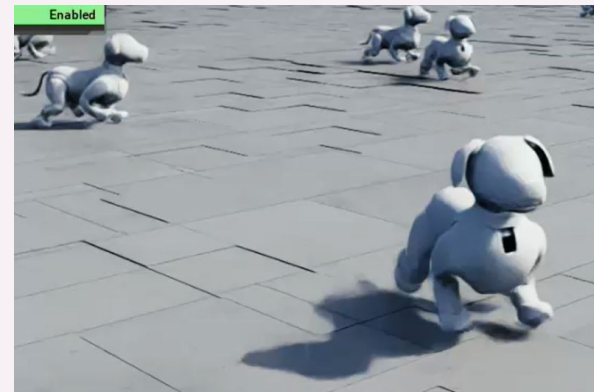
- Joint Encoder
- IMU Sensor
- Foot Contact Switch



Action

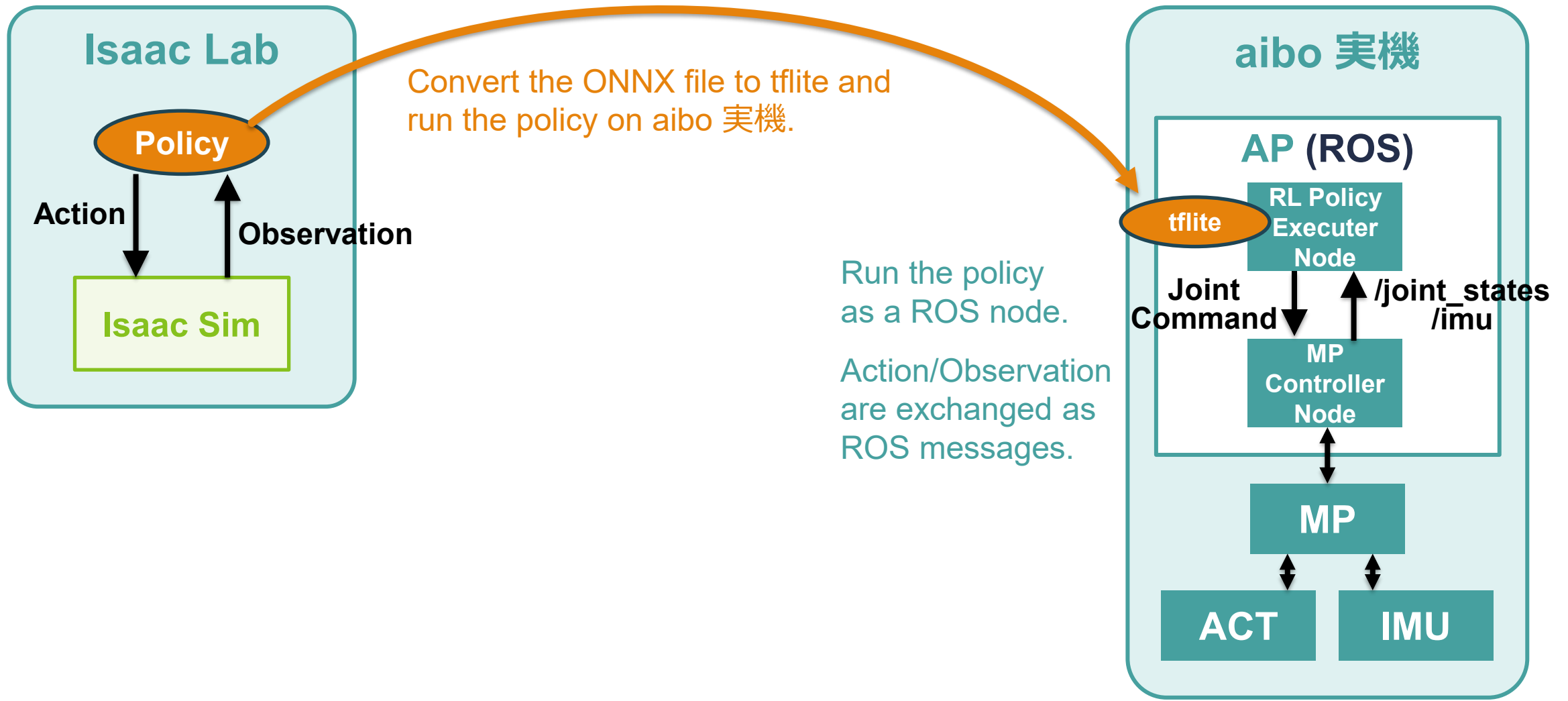
- Target Joint Position
- Target Joint PD Gain^[1]

Agent & Environment

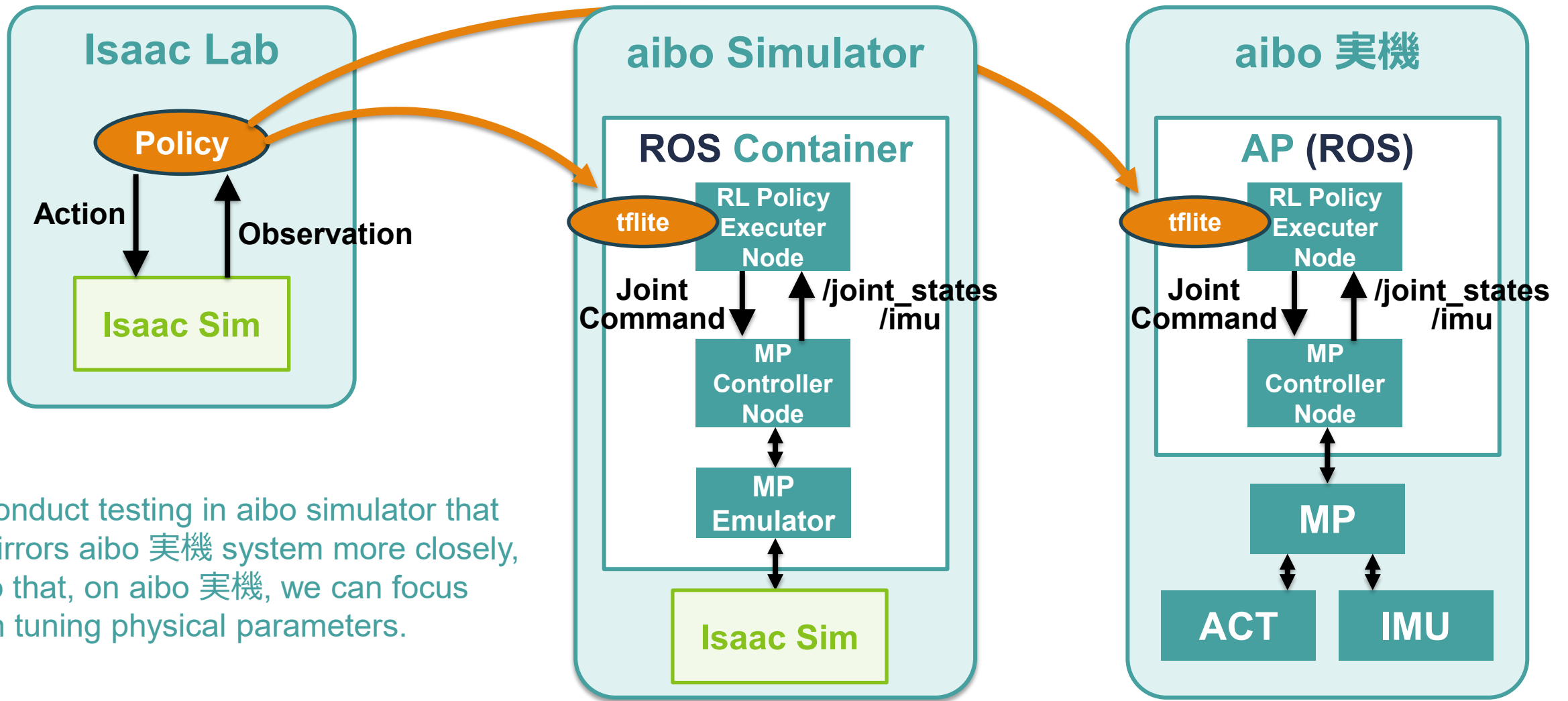


[1] R. Watanabe et al., "Learning Quiet Walking for a Small Home Robot," in *Proc. of ICRA 2025*, 2025.

Motion Reinforcement Learning



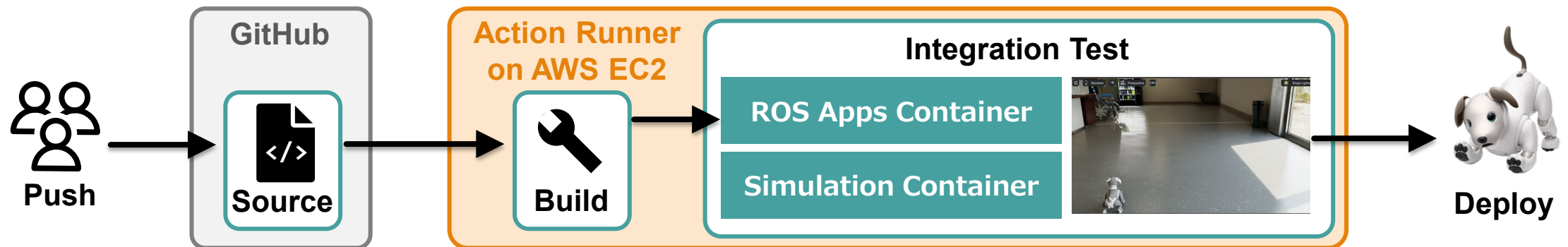
Motion Reinforcement Learning



Conduct testing in aibo simulator that mirrors aibo 実機 system more closely, so that, on aibo 実機, we can focus on tuning physical parameters.

Conclusion

- Thanks to high-quality rendering and physics simulation, Isaac Sim enables end-to-end simulation, from perception through motion control.
- Isaac Sim is highly compatible with Python and ROS, and is well-suited for simulation of existing ROS robot applications.
- With support for container environments, it integrates well with cloud computing and CI/CD pipelines.



Let's explore what's next with aibo.
Bring your research or ideas to life with aibo.



SONY

SONYはソニーグループ株式会社の登録商標または商標です。

各ソニー製品の商品名・サービス名はソニーグループ株式会社またはグループ各社の登録商標または商標です。その他の製品および会社名は、各社の商号、登録商標または商標です。