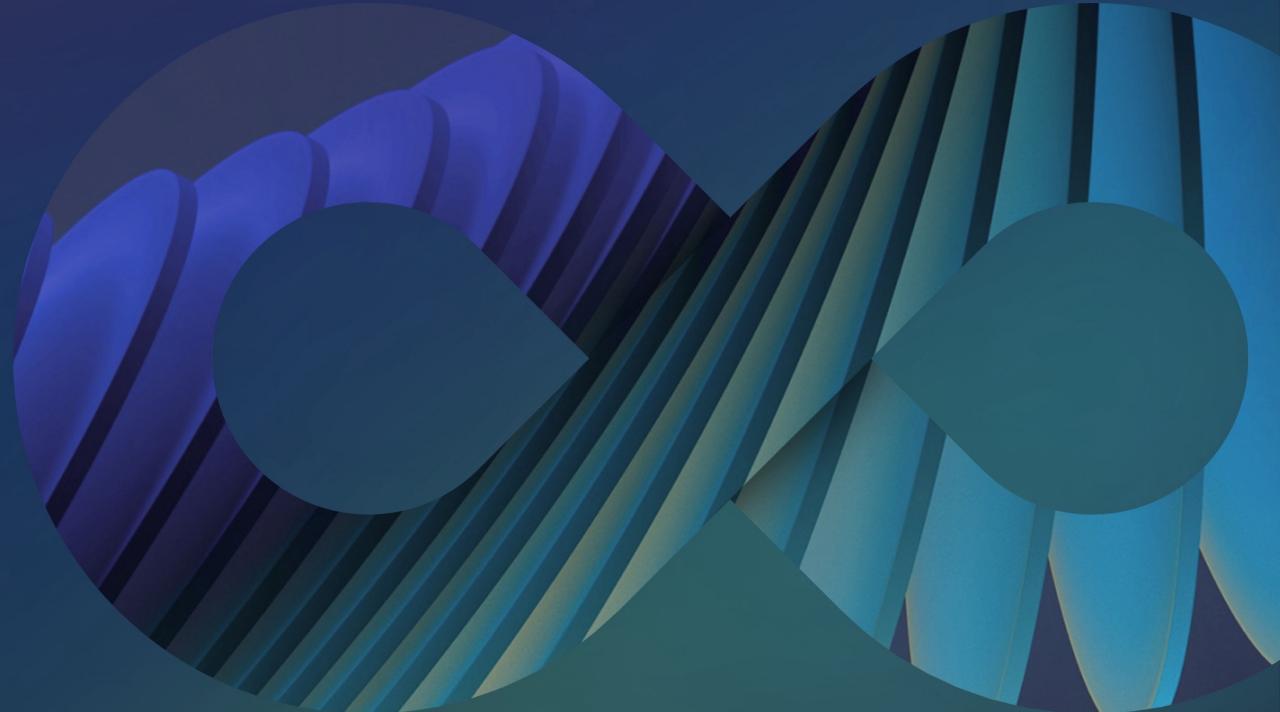


Interactable Digital Twins for Robots via WoT

Ege Korkan (Siemens AG)

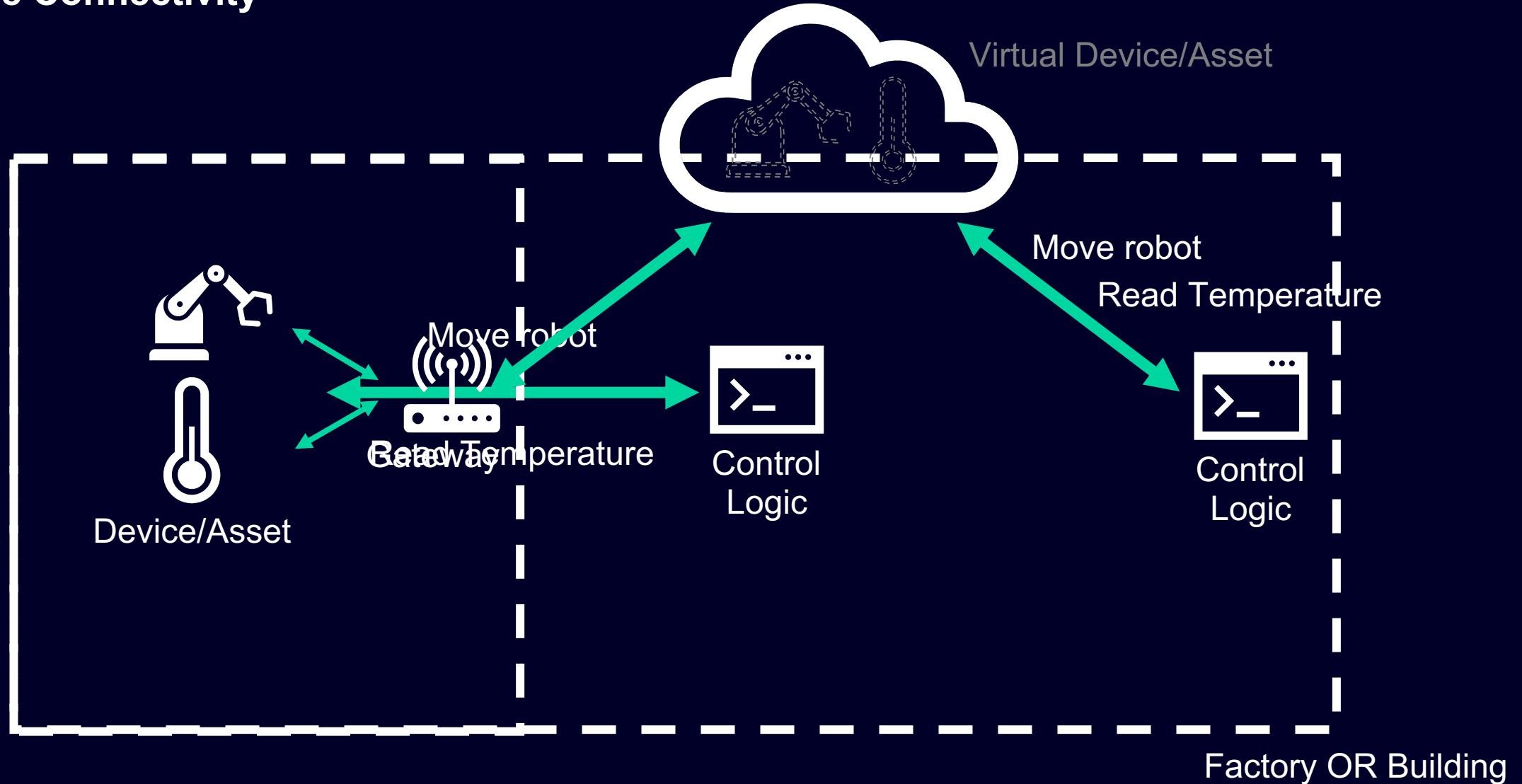
Fady Salama (Technical University of Munich)



What is Device Connectivity in the first place?

At least for this presentation ;)

Device Connectivity



Simulation Software in General

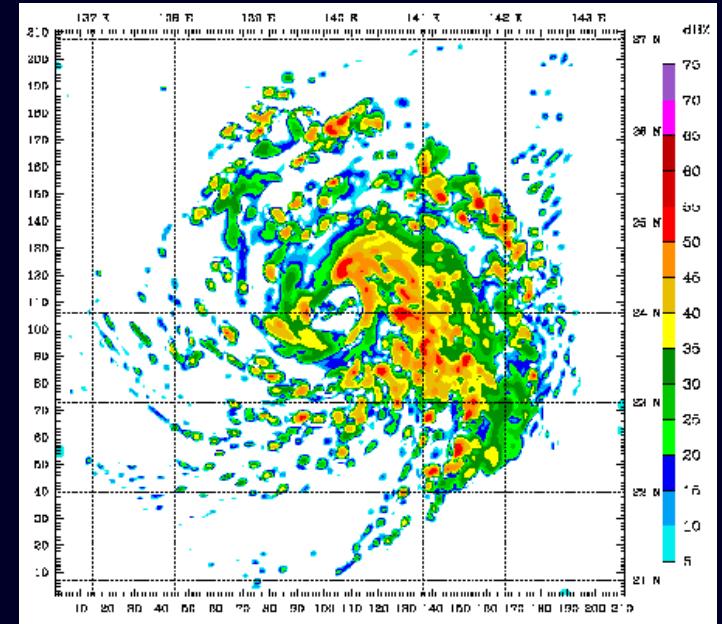
What is common

Simulations ... or are they now called Digital Twins?

Goal: predict the behaviour of, or the outcome of, a real-world or physical system

Typical steps:

1. Model a System
2. Run Simulation
3. Get Results and compare with real life



Important to remember in this presentation:

- **We do not want to reinvent this, nor do we have the knowledge**

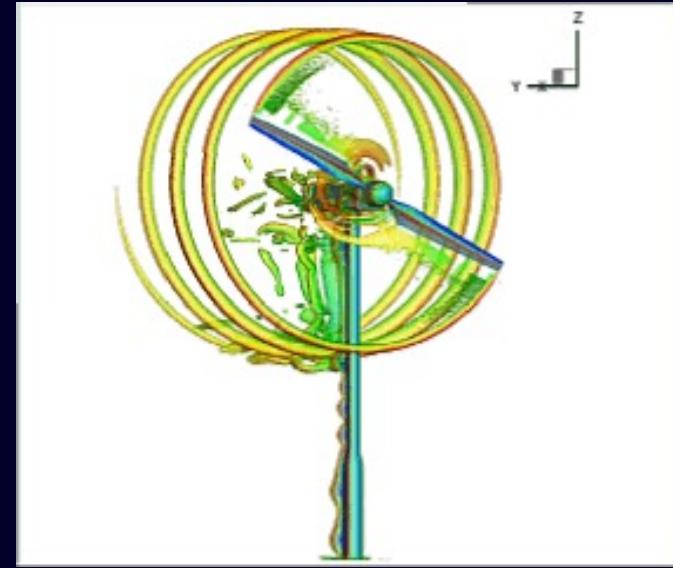


SIEMENS

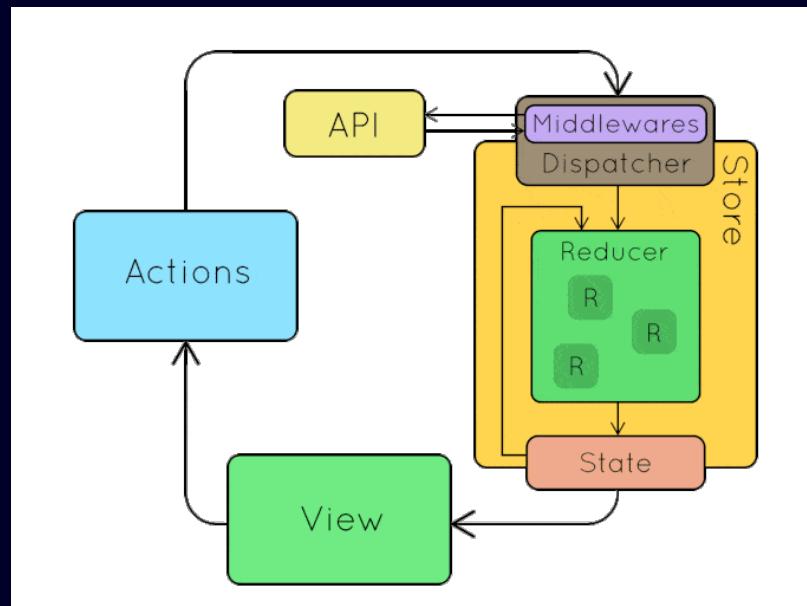
Simulations ... or are they now called Digital Twins (DTs)?

Depending on desired fidelity or abstraction, they can be:

- real time (or faster) or → Our scope
- (much) slower than real time → not suitable for interactable DTs
more for design stage with optimizations

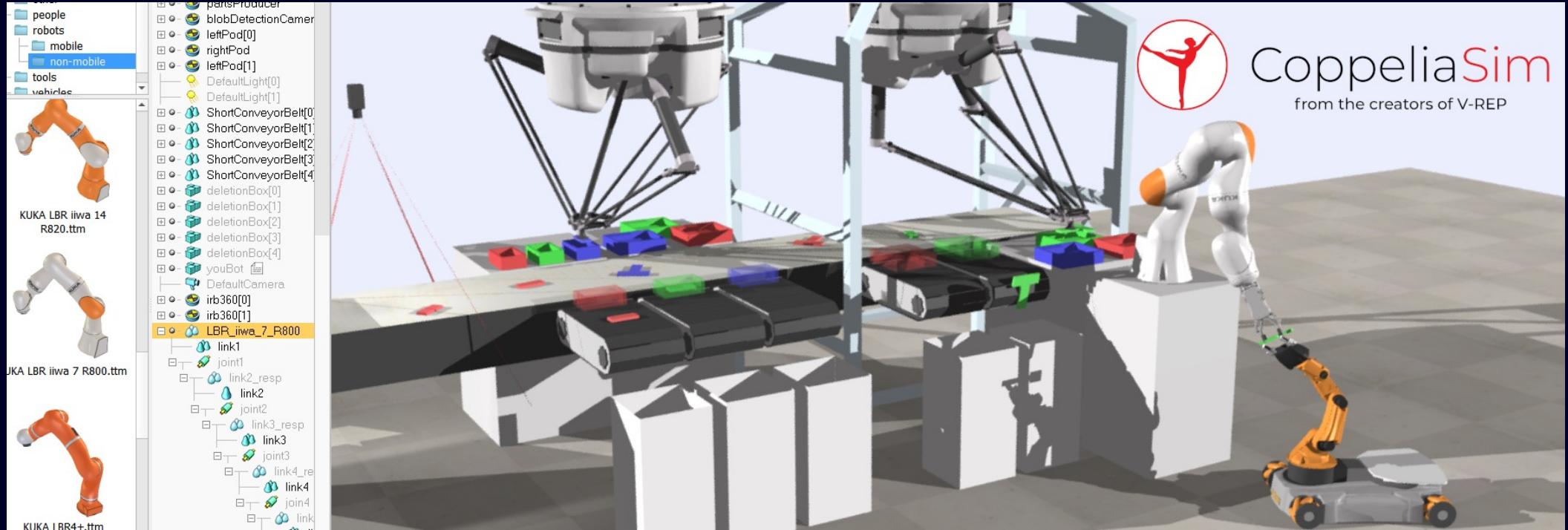


Software running, typically a desktop application, optionally using resources of a high-performance compute cluster.



SIEMENS

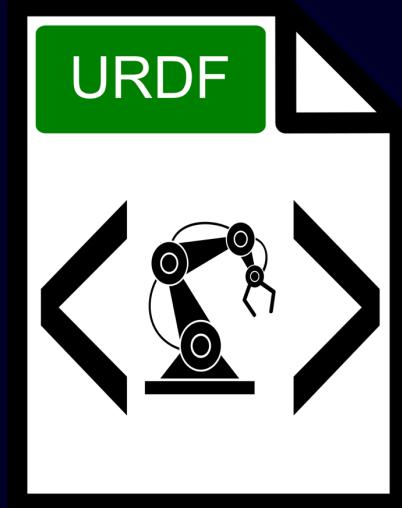
Meal of the Day: Robotics Simulations



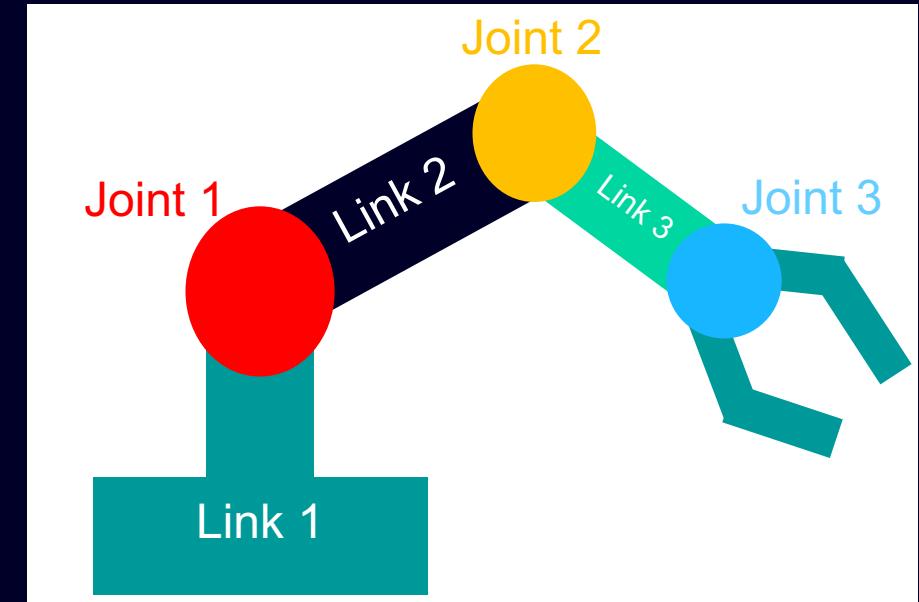
From <https://www.coppeliarobotics.com/>

If we can get Thing Models (TMs) from manufacturers, what about robot models?

Also from them via Universal Robot Description Format (URDF)



- XML file format
- Describes Robots using a hierarchy of links and joints
- Can describe geometry and physical attributes
- STL model (3D CAD) of the robot can be linked
- Can be used to exchange robot models



Many companies such as ABB, KUKA, FANUC, STAUBLI make URDF of their robots open source at <https://github.com/ros-industrial> and <http://wiki.ros.org/urdf/Examples>

RobWoT

Our Approach for Digital Twins for Robots
and maybe more

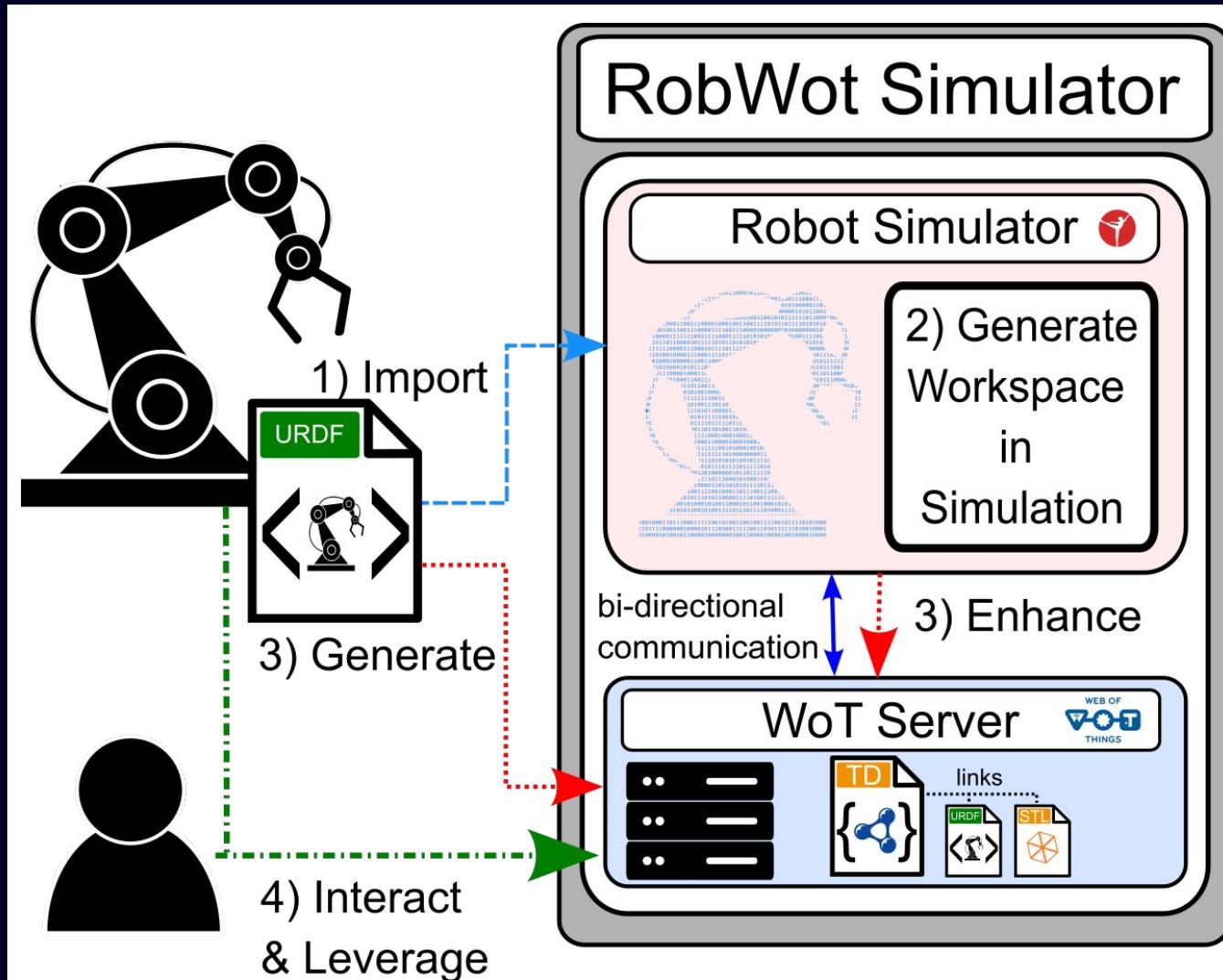
Straight to Demo

We will see:

- A real-time video stream of lab setup in TU Munich
- A digital twin of it working in my laptop within Coppelia Sim

The same function calls are executed on both real and virtual devices

How does it work?

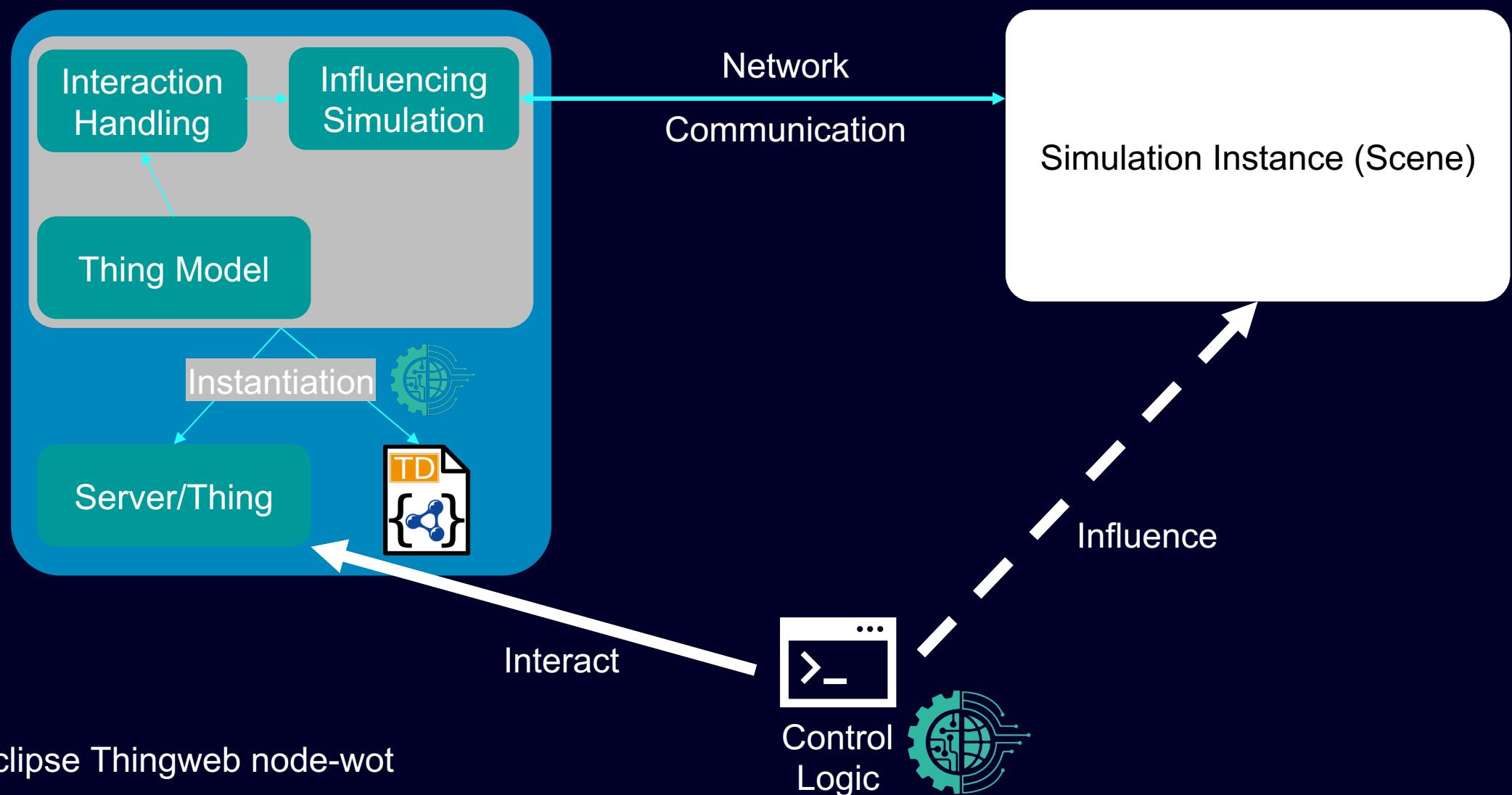


How can a developer use it?

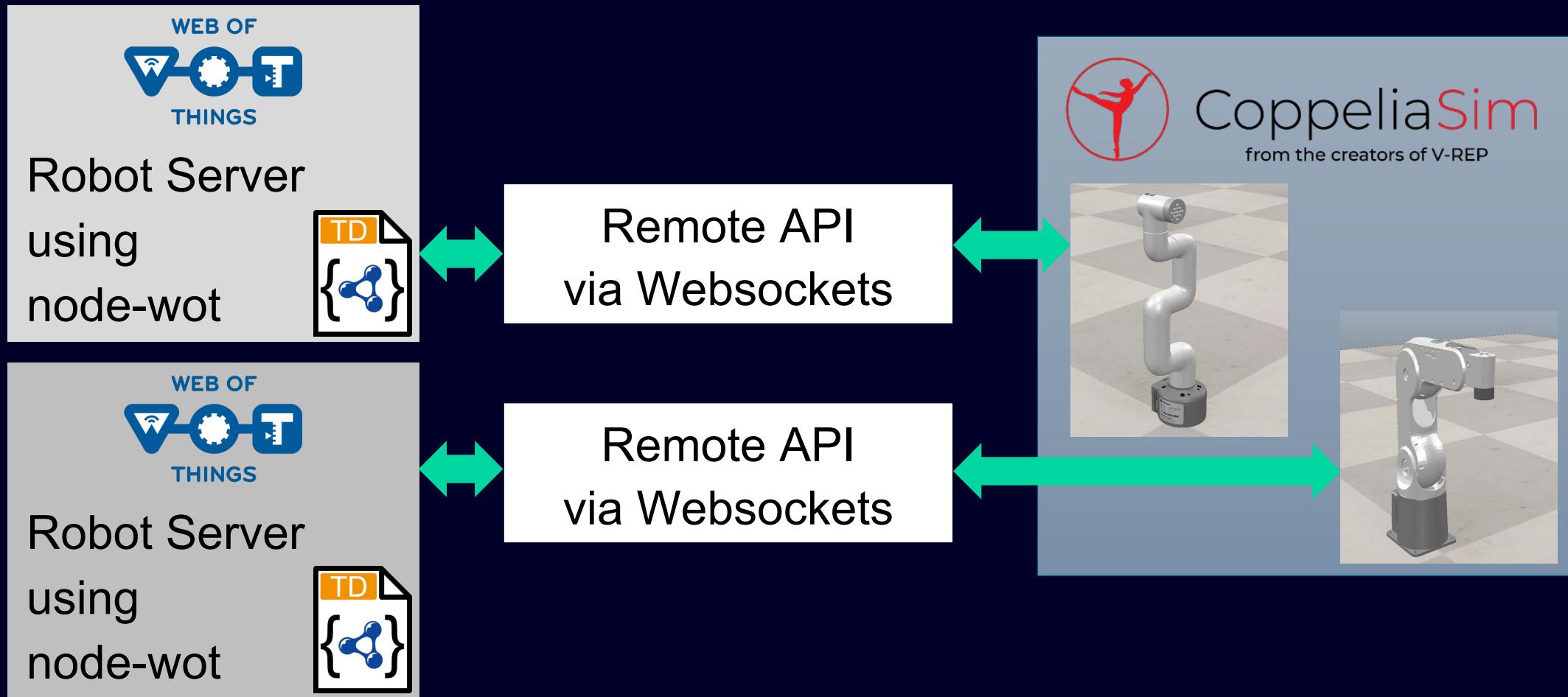
1. Get the TM (from manufacturer)
 2. Get the URDF (with STL) (from manufacturer)
 3. Reuse a simulation software (here Coppelia Sim)
 4. Bridge the TM to the simulation logic via code
 5. Start the program to generate the TD and make the simulation interactable
-
- No development effort is needed

Interact with the real device or simulation via very similar TDs so that the control logic stays the same

Components of a Simulated Thing



Interfacing with the CoppeliaSim



Code Example of a Conveyor Belt

RobWoT is open-source at <https://github.com/tum-esi/RobWoT> so you can try it yourself

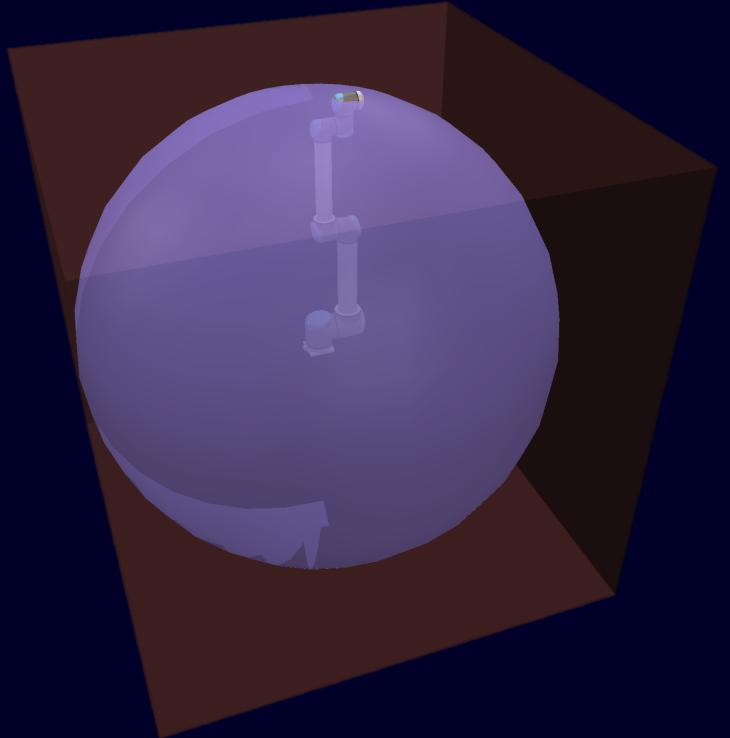
Let's look at <https://github.com/tum-esi/RobWoT/blob/main/SimulationThings/ConveyorBelt/ConveyorBelt.ts>

Note: It is also published as a paper at [WF-IoT23](#)

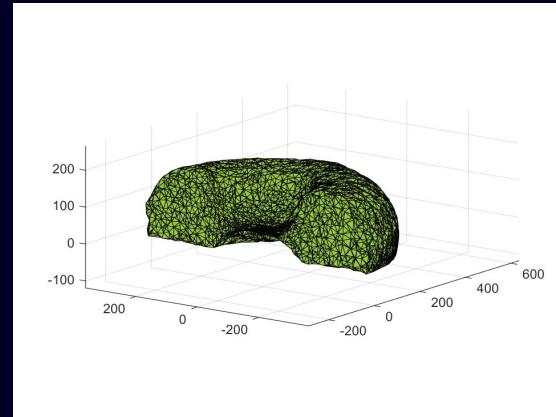
Side Contribution: Making Robots Usable in WoT

Robots – very special devices

- TDs are cannot describe robot arms perfectly:
 - Bad at describing the available workspace of robots
 - Even worse if the robot is obstructed by other objects in the environment



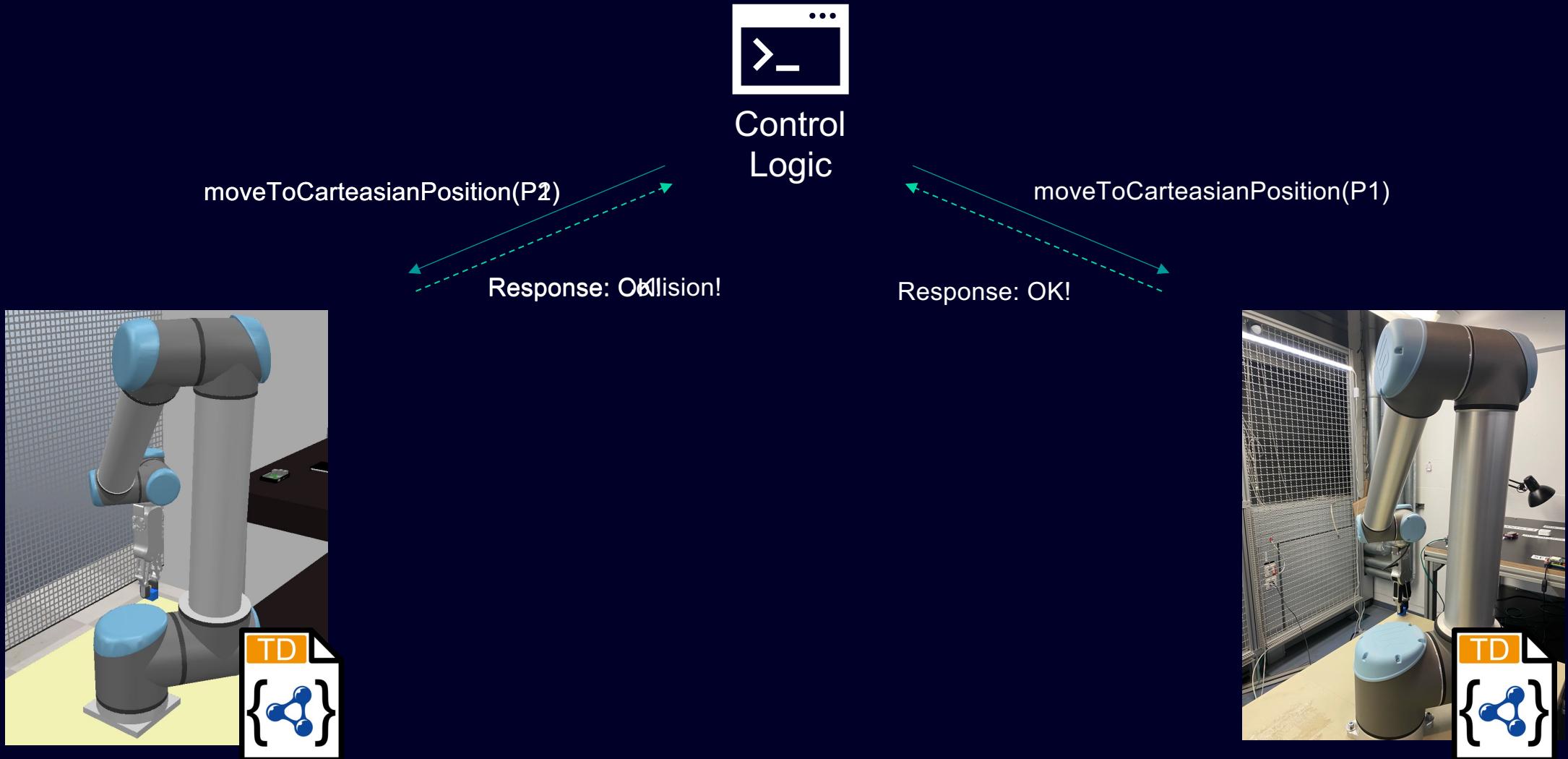
Approach



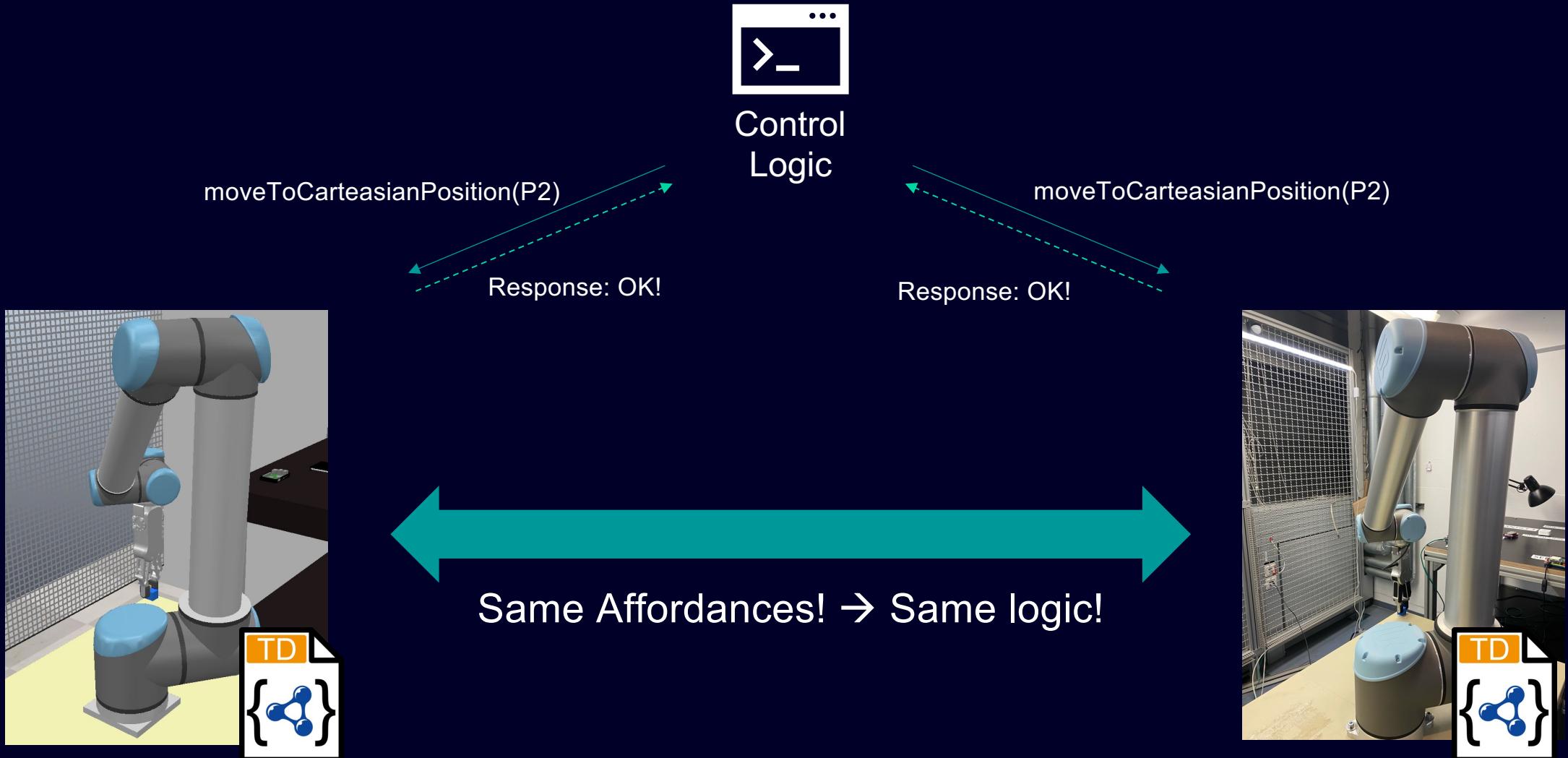
1. We calculate the available workspace with different movements
2. Available workspace can be understood by advanced consumers OR
3. The simulation can be kept in the loop to be sure that desired movement can be achieved

What do we enable?

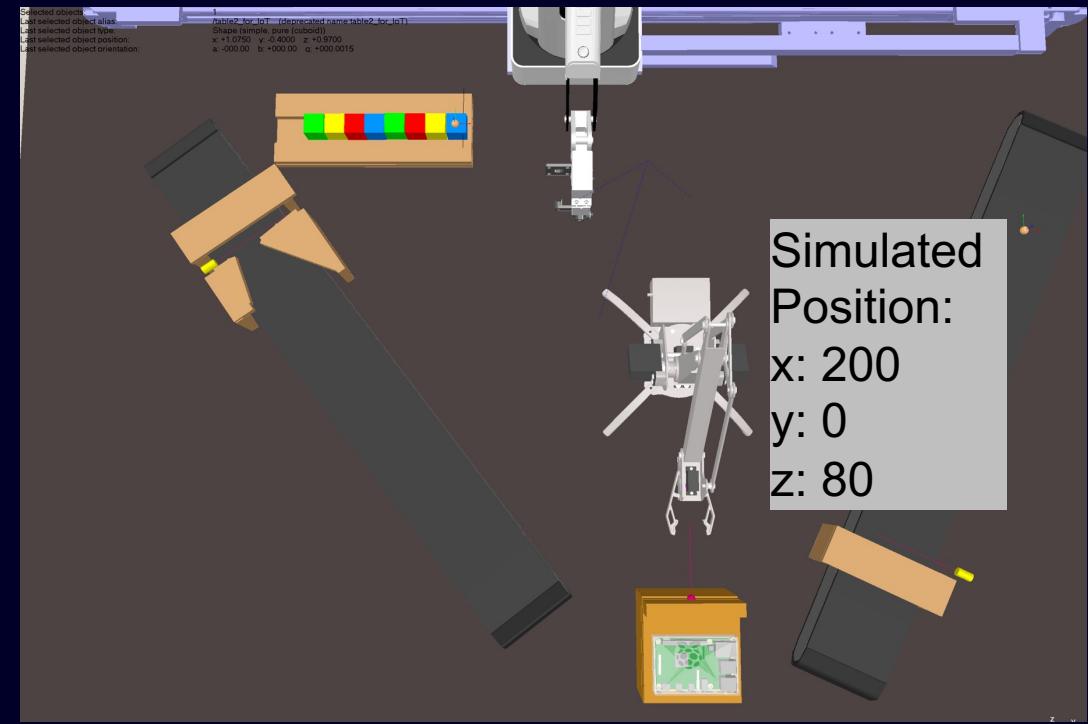
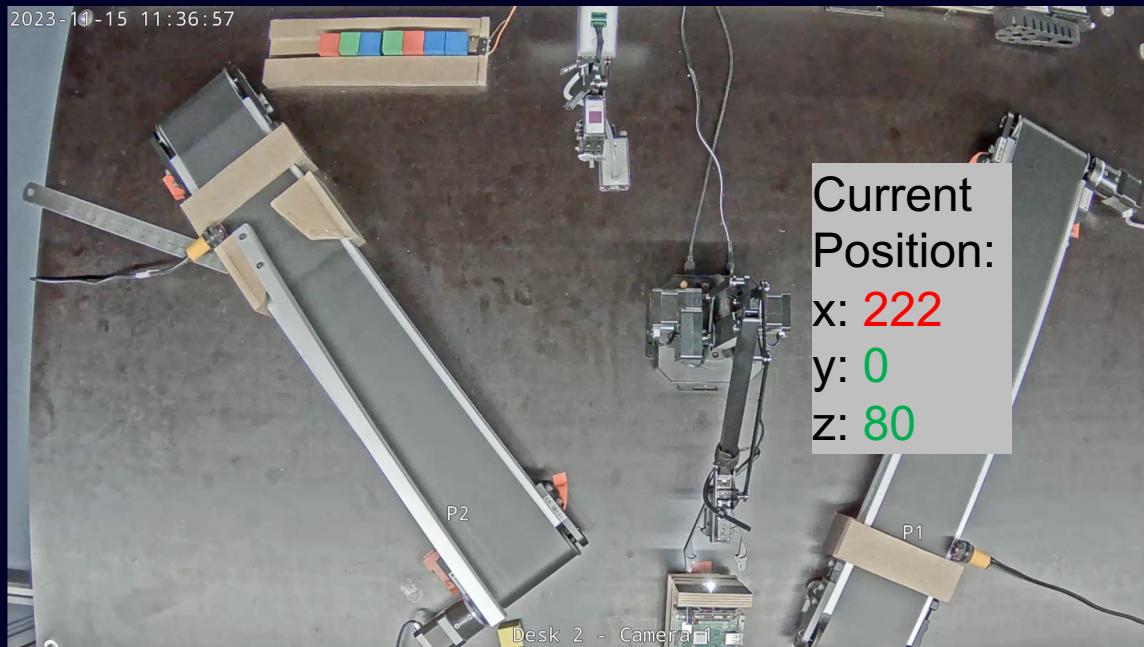
Dry run or Simulation in the Loop



Dry run or Simulation in the Loop



Compare Reality with Plans (in the virtual world)



Add connectivity to your simulation software

With this approach, you can embed a flexible, interoperable and standardized connectivity solution to enable digital twin applications.

Hopefully, you can be inspired from the approach and implement it your own way to fit your simulation needs.

One more *Thing*

A more complicated (and error prone) demo

Similar to previous demo but more sequential interactions with the robots!

We put it here since there is more risk of something going wrong 😊

Also, the timing is not perfect but we are working on it.

Contact

Ege Korkan – Siemens AG
ege.korkan@siemens.com

Fady Salama – Technical University of Munich
fady.salama@tum.de