



# Using Rust for Playing Soccer With Robots at RoboCup



Hendrik Sieck  
**HULKs**



# RoboCup

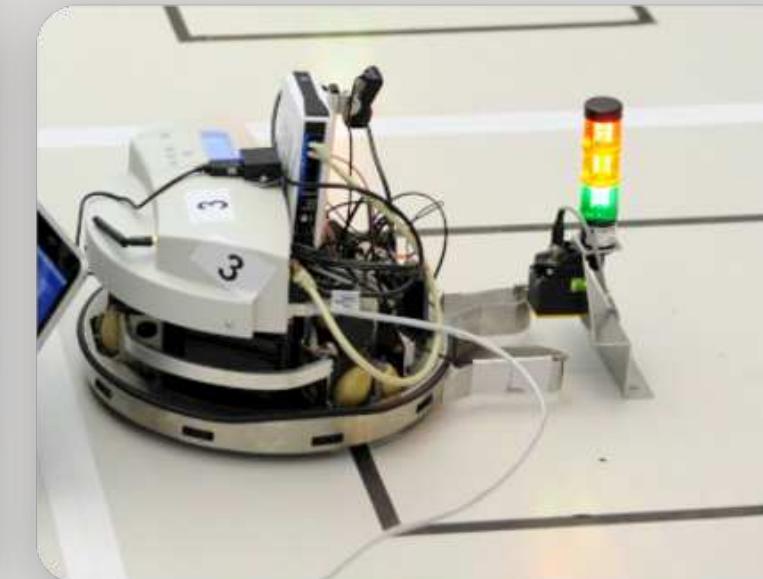
Rescue



At Home



At Work



Logistics



Soccer



Simulation



Small Size



Mid Size



Humanoid

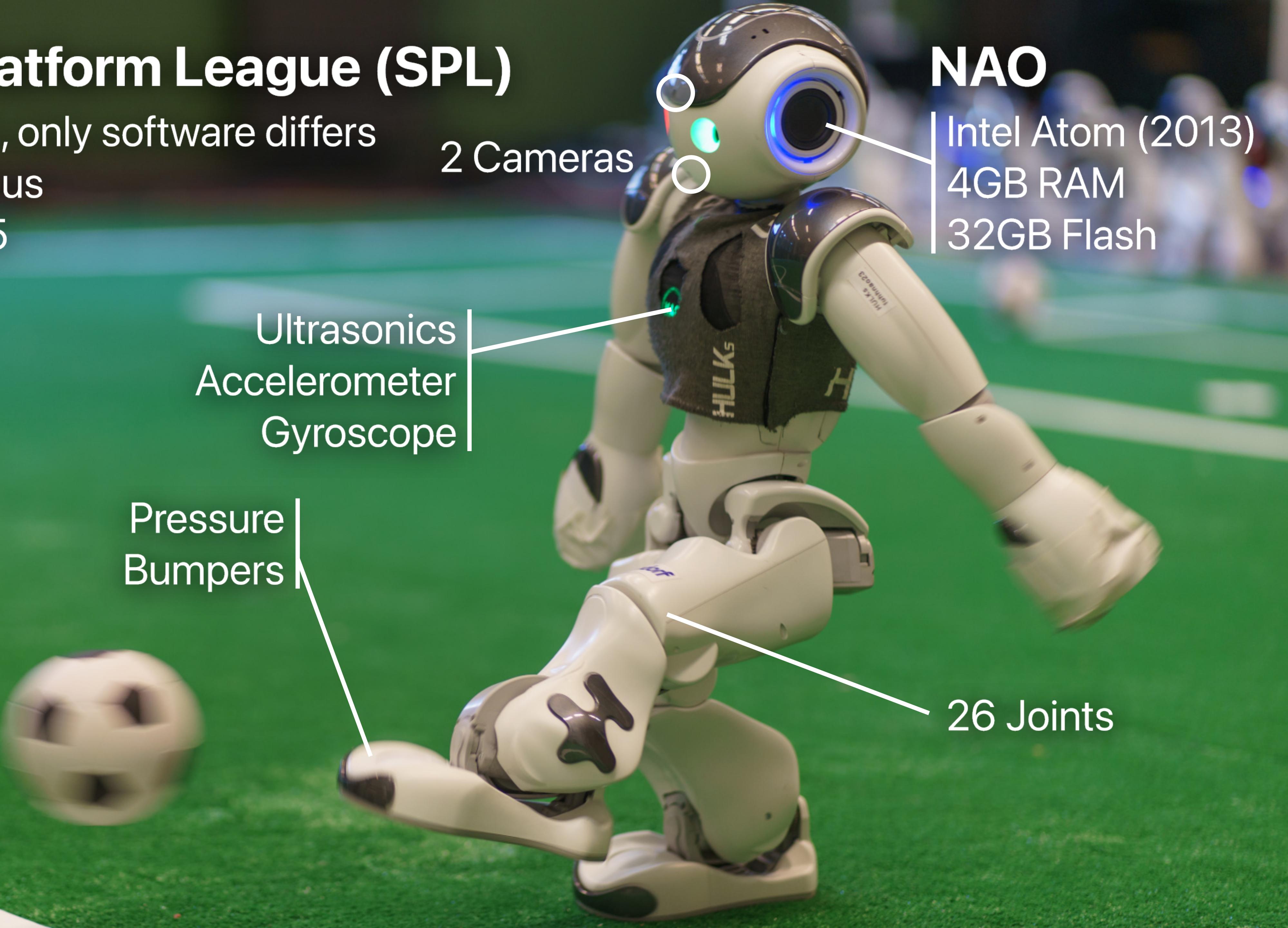


Standard Platform

Goal: Robots vs. Humans in 2050

# Standard Platform League (SPL)

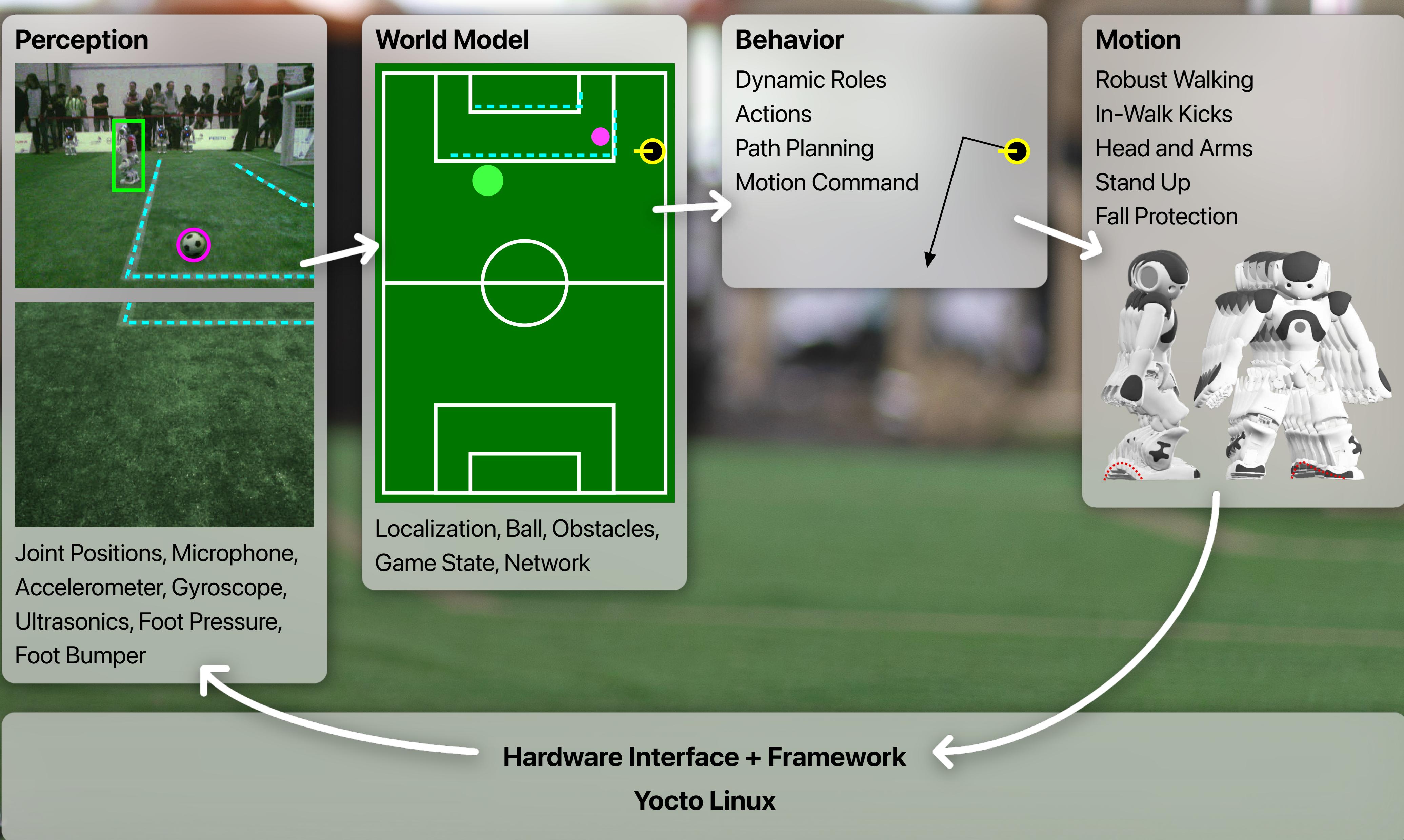
- Identical robots, only software differs
- Fully autonomous
- 7 vs. 7 or 5 vs. 5



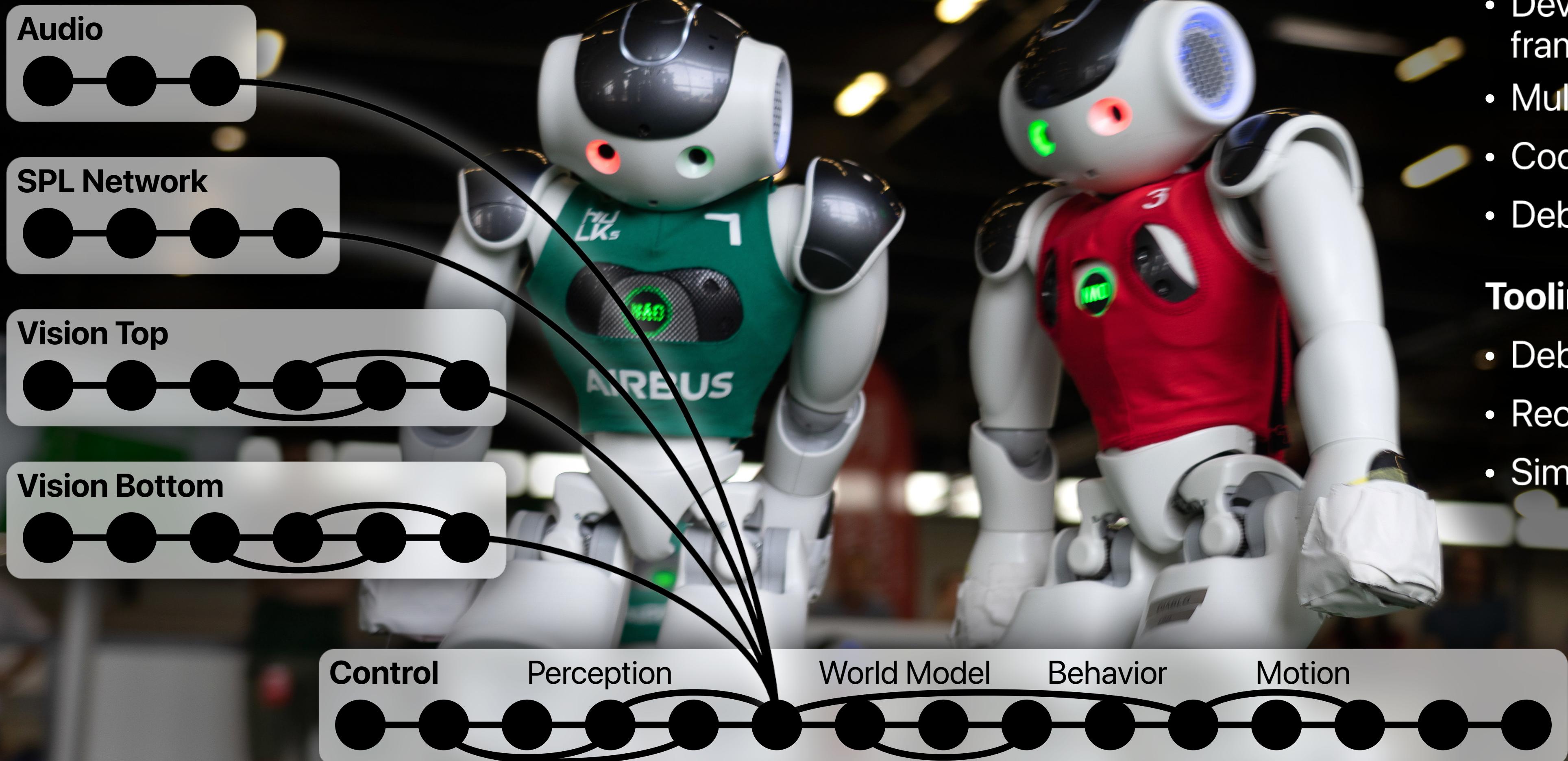
# Team HULKs



# Robot Control Software



# Software Design



## Framework

- Developer writes nodes, framework executes
- Multi-threading
- Code generation
- Debugging server

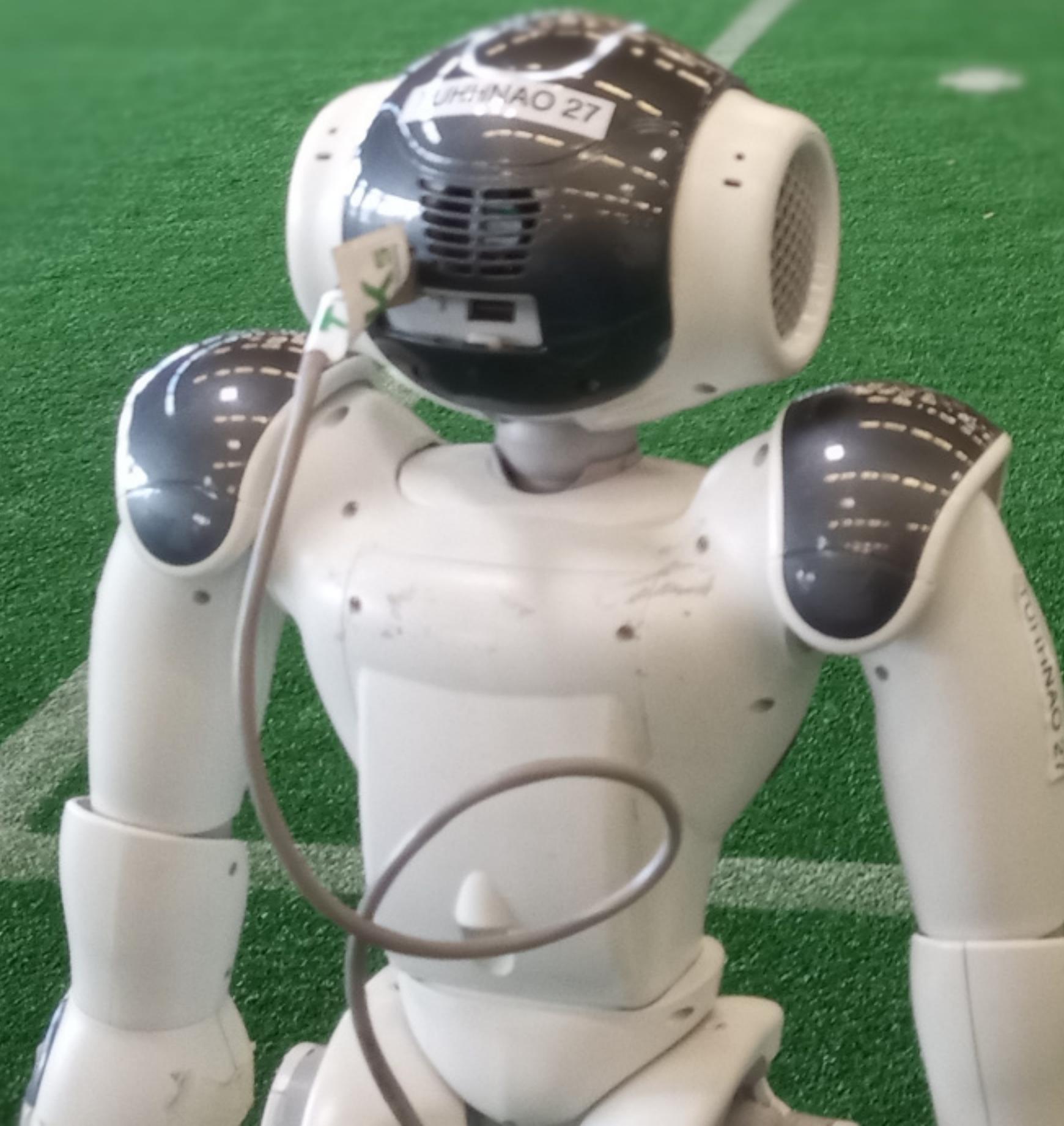
## Tooling

- Debugging clients
- Recording & Replay
- Simulators

# Transitioning from C++ to Rust

## Why?

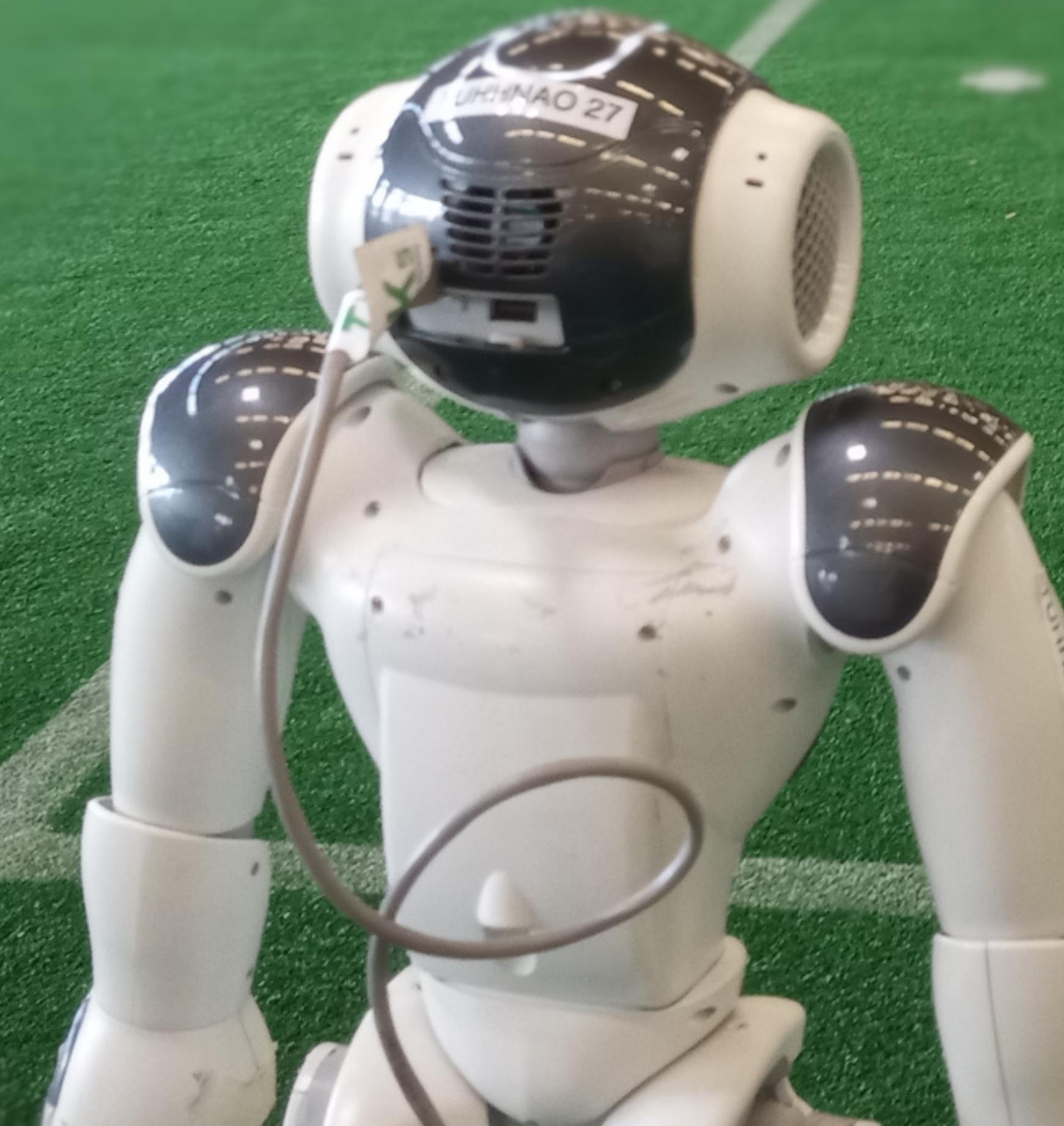
- Decrease technical debt, rethink main concepts
- Knowledge transfer, newbie motivation
- Learn a new language
- Fun ;)



# Transitioning from C++ to Rust

## Why?

- Decrease technical debt, rethink main concepts
- Knowledge transfer, newbie motivation
- Learn a new language
- Fun ;)



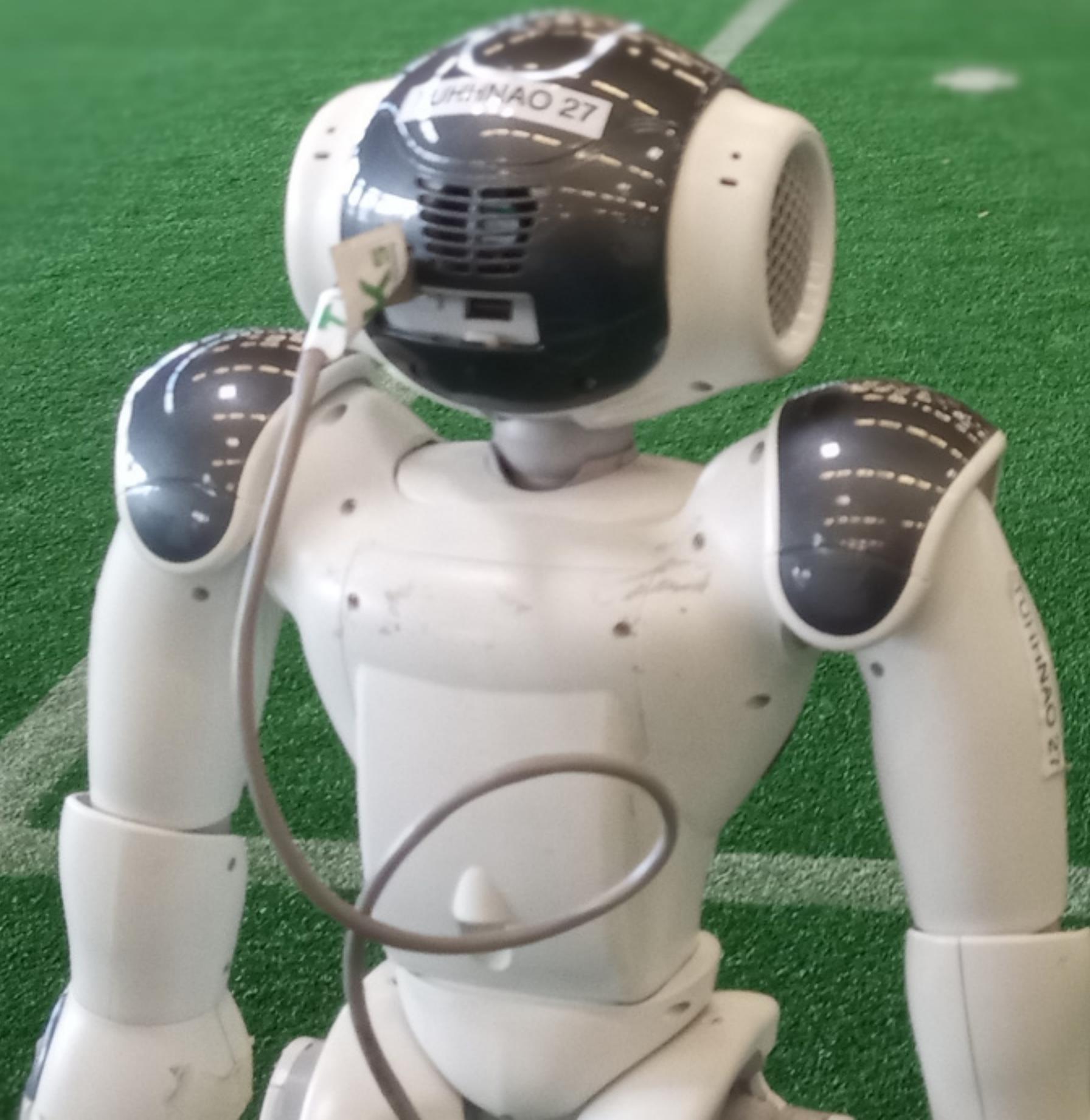
## Timeline

- November 2021: Start of concept and design
- After 2 months: Working framework prototype
- After 5 months: Technical demonstration at GORE  
Scored 3 goals! 1 in opponent goal ;)
- After 8 months: Quarter finals reached at RoboCup  
Scored 17 goals!
- Present: Further iterations of framework

# Transitioning from C++ to Rust

## Why?

- Decrease technical debt, rethink main concepts
- Knowledge transfer, newbie motivation
- Learn a new language
- Fun ;)



## Timeline

- November 2021: Start of concept and design
- After 2 months: Working framework prototype
- After 5 months: Technical demonstration at GORE  
Scored 3 goals! 1 in opponent goal ;)
- After 8 months: Quarter finals reached at RoboCup  
Scored 17 goals!
- Present: Further iterations of framework

## Learnings

- Compiler + tooling → fearless change, velocity, risk taking
- Language and standard library has richer feature set
- Development feels longer, but then it just works
- No SEGFAULT infrastructure, high prototype quality
- Focus on domain and real problems, not optimization
- To infinity and beyond: GUI, Web, Embedded, ...



# Current State & Tools

- Newbie Education** (Rust Introduction, Pairing)
- Monorepo + Workspace Dependencies** (cargo)
- Yocto Integration** (depp, cargo-bitbake)
- Additional Tooling** (clap, indicatif)
- CI Setup** (test, clippy, rustfmt)
- UI Tools** (egui, tungstenite + Web)
- Profiling** (VTune, pprof)
- Linear Algebra Libraries** (nalgebra)
- Coordinate Systems**
- Error Handling** (anyhow, eyre, color-eyre)
- Serialization** (serde)
- Machine Learning** (CompiledNN, OpenVINO)
- Interfacing to C++ and other stuff** (bindgen, mlua)



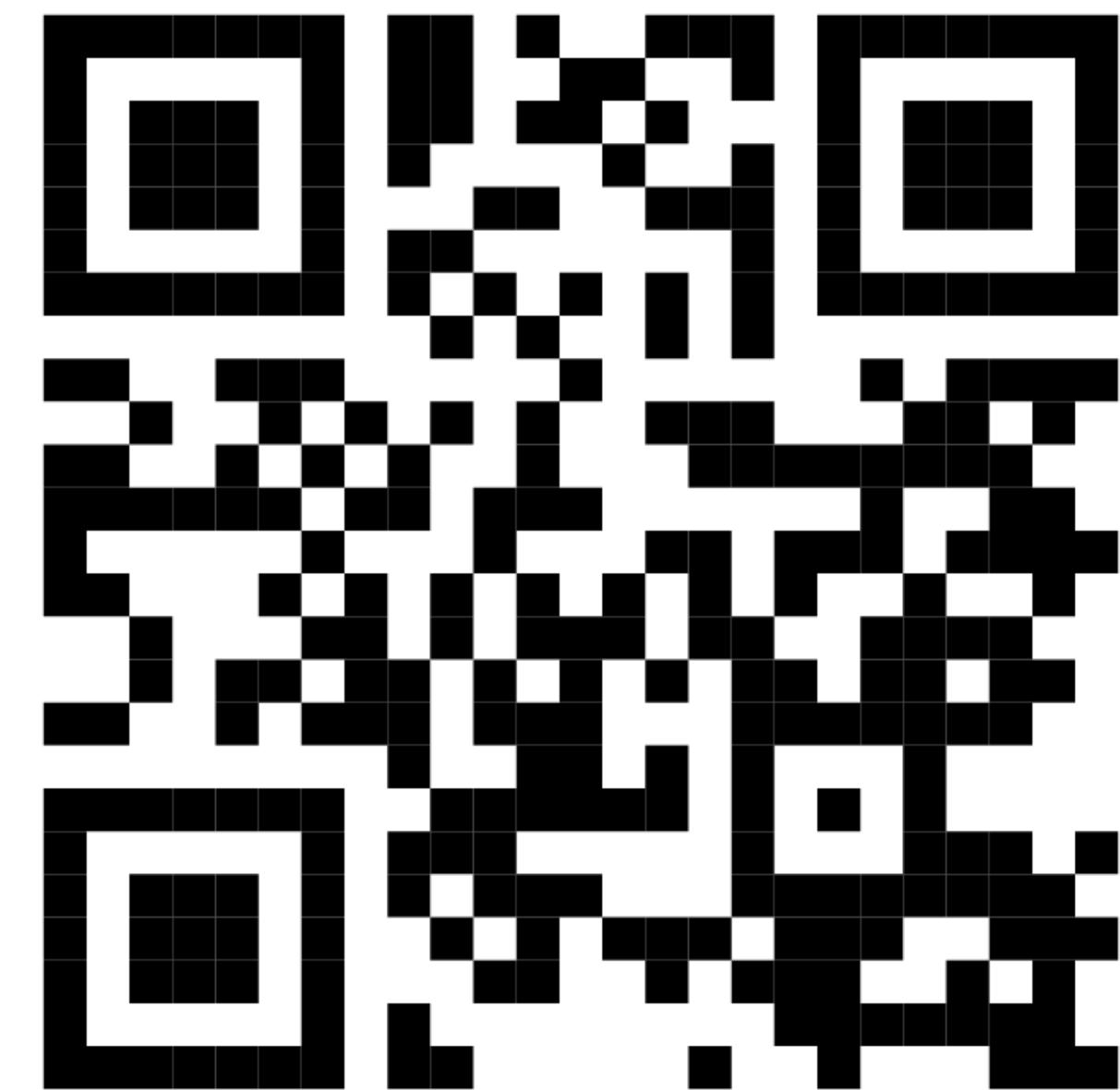
# Live Demo

# HULKs Code



<https://github.com/HULKs/hulk>

# Slides and other Talks



<https://github.com/h3ndrk/talks>



Hendrik Sieck  
**HULKs**  
zühlke empowering ideas