Building dependable products with ROS

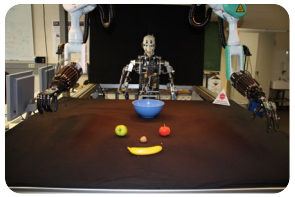
Dr.-Ing Ingo Lütkebohle
Bosch Research

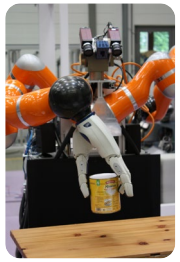
For ROSCon JP, September 26th, 2023

BOSCH

Dr.-Ing. Ingo Lütkebohle

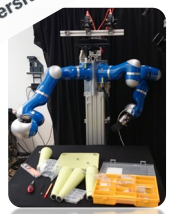
UNIVERSITÄT BIELEFELD



2005 – 2013

Universität Stuttgart



2010 & 2013

BOSCH



ROS

2014-

From system integration to dependability
From HRI to embedded systems

BOSCH

About Bosch
Our business sectors

Mobility Solutions

Industrial Technology

Energy and Building Technology

Consumer Goods

Ingo Lütkebohle | 2023-08-29

BOSCH

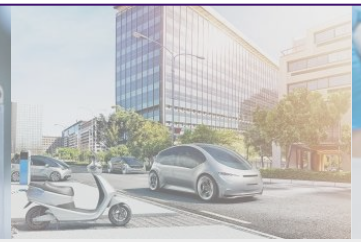
Bosch Research Focus Areas

Autonomous Intelligent Driving (AID)

Sensing, perception, prediction, planning
Systems & infrastructure for L2-L4

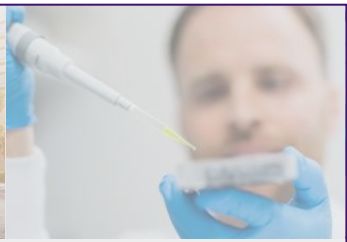
Chemical Energy Converters (CEC)

Hydrogen electrolysis
PEM fuel cell for EVs
Solid Oxide Fuel Cell



Electrified Mobility and Systems (EMY)

Electric drives
Power electronics
Integrated electrified products



Healthcare Solutions (HCS)

Point-of-care lab diagnostics
Liquid biopsy
Next gen sequencing

IoT @ Life (IOT)

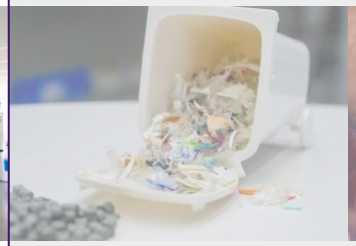
Enabling AIoT for mobile, residential, tools and multi-domain applications

Production Systems (PRS)

Production technologies
AI in production
Internet of production

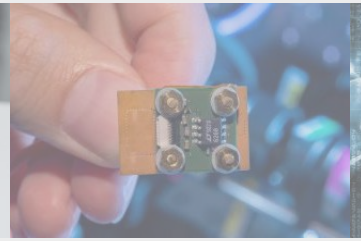
Robotic Systems (ROB)

Consumer robots
Industrial service robots
Industrial robot arms



Sustainability (SST)

Sustainability big picture
Climate change mitigation
Circular economy
Carbon dioxide removal



Smart Sensors & HW Systems (SSY)

MEMS
Quantum sensors
HMI technology
Smart systems
Embedded AI hardware

Artificial Intelligence Methods (AIM)

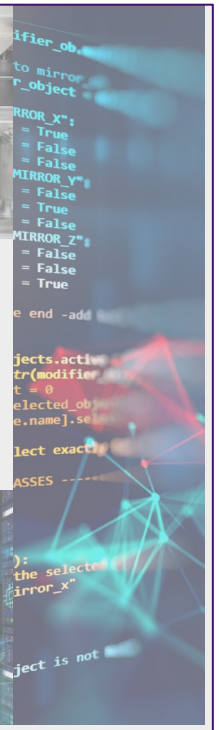
AI data loop enablers
Natural language processing
Computer vision
AI method incubator

Information and Communication Tech (ICT)

Software & systems engineering
Distributed infrastructure
Intelligent functions & services

Modeling, Simulation, Optimization (MSO)

Sustainable engineering
Virtual product design
Virtual validation
Use phase monitoring

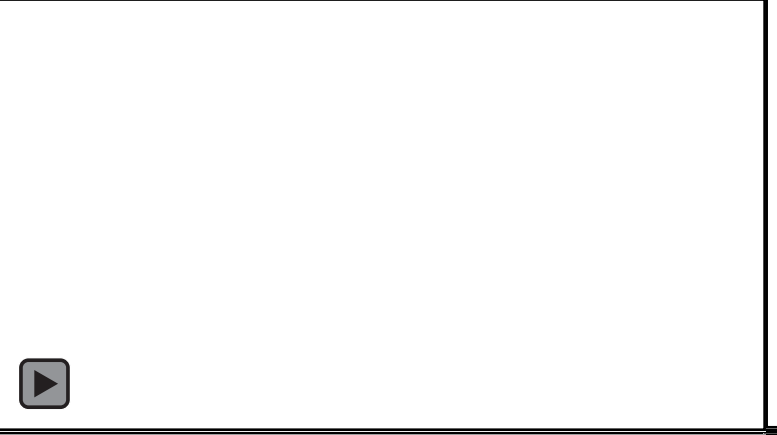
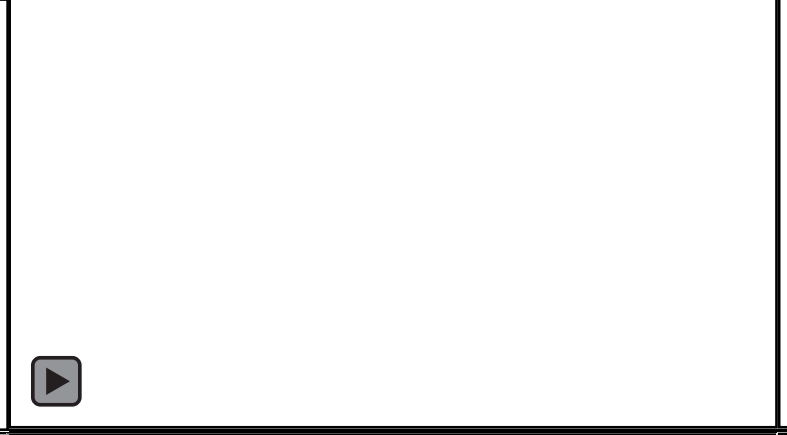
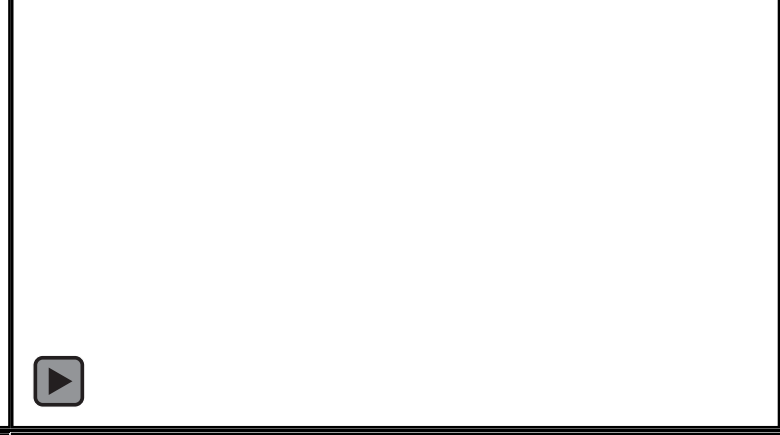
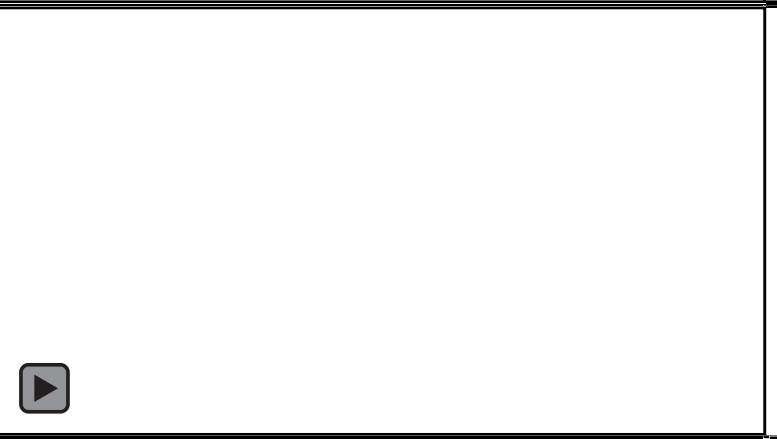
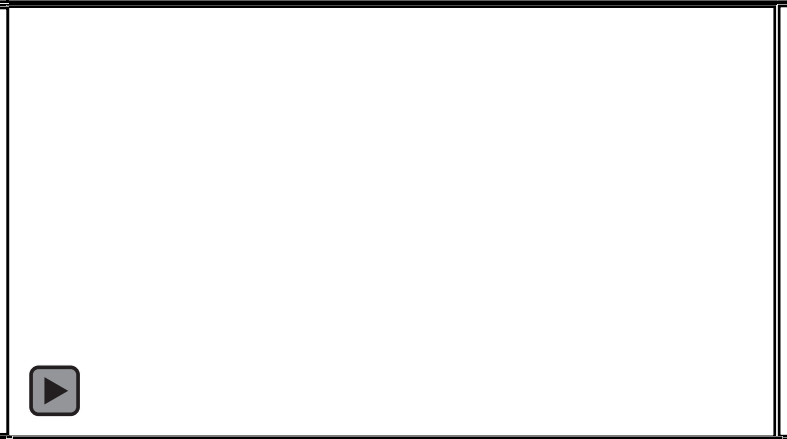


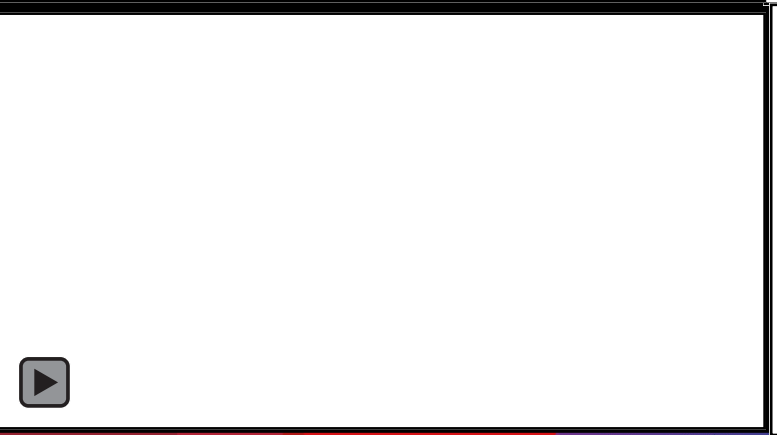
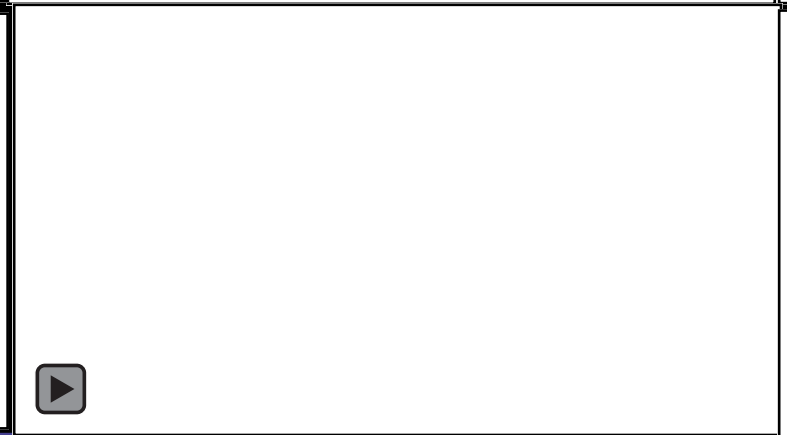
Industrialization of AI & SW (BIS)

Providing mature AI & SW artefacts

01

Robotics at Bosch


Bosch's 12 Year Journey with ROS

Bosch participates in PR2 beta program

Founding member of ROS-I Europe

Bosch sponsors development of ROS 2

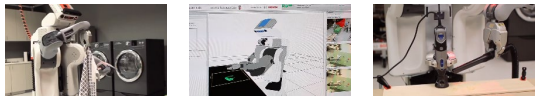
Founding member of ROS TSC

EU project micro-ROS

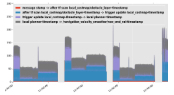
EU ITP MROS

EU project CONVINCE

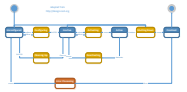
2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022



Tools and basic algorithms from PR2 beta program

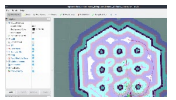


Tracepoints

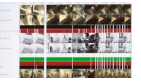


Node lifecycle

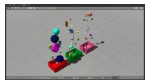
UUV simulator



rviz2



rosbag2



pcg_gazebo

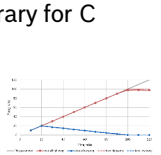
System Modes

fmi Functional Mock-Up Interface

fmi_adapter

Middleware adapter

Client library for C



Refined Executor

ros2_control

Diagnostics

From a small research team to hundreds of developers using ROS

Ingo Lütkebohle | 2023-08-29

BOSCH

Steering of Our ROS Strategy and Infrastructure

Consumer robotics

Commercial vehicles & off-road

Industrial robotics

Bosch Robotics Middleware TSC

Professional service robotics

Inner source

Open source

Ingo Lütkebohle | 2023-08-29

BOSCH

02

Why ROS?

Academic Frameworks before ROS

Year	Framework(s)	Citations
1994	TCA	~500
1997	DAMN	~700
1998	SmartSoft	<20
1999	ISR	<10
2000	OSCAR	<20
2001	OROCOS MCA	>800 ~100
2002	RoboMote	>500
2003	Player/Stage CARMEN ROCI	>2000 ~300 ~50
2006	YARP MOOS ORCA	>800 ~250 ~200
2007	STAIR	~150
2009	**ROS**	**>11000**
2010	LCM Fawkes	~400 ~80

- Industrial frameworks largely robot-specific
  → Academic Open Source Frameworks
- These have many similarities on the *plumbing* level

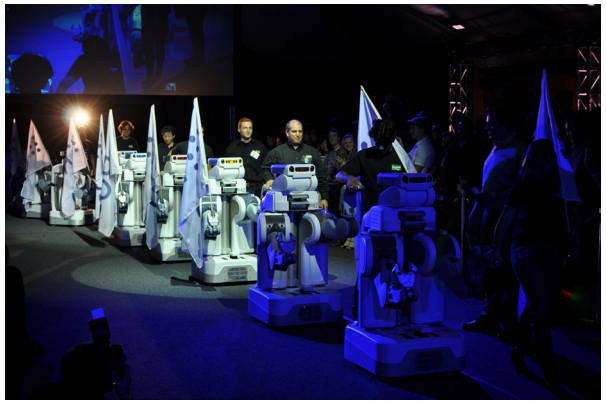
ROS Success Factors

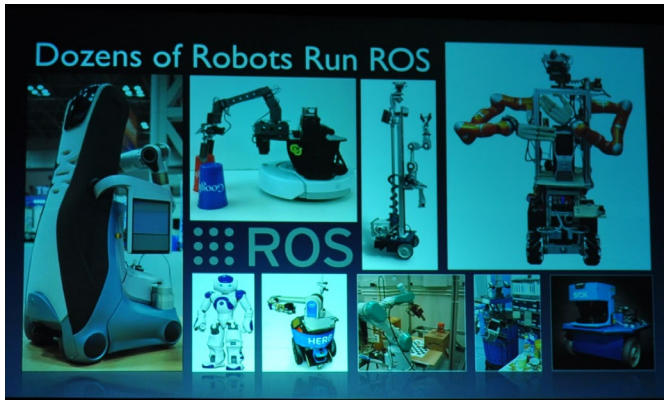
Which of these is most important?

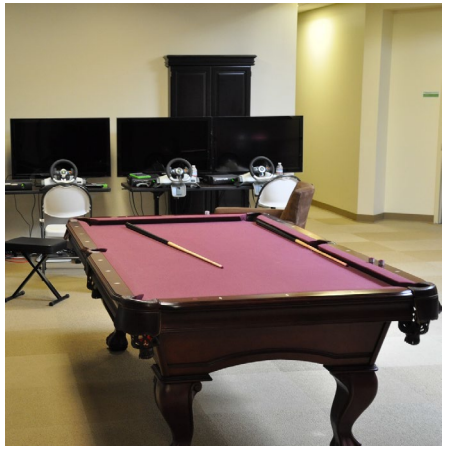
Image Source: https://www.ros.org/blog/ecosystem/

BOSCH

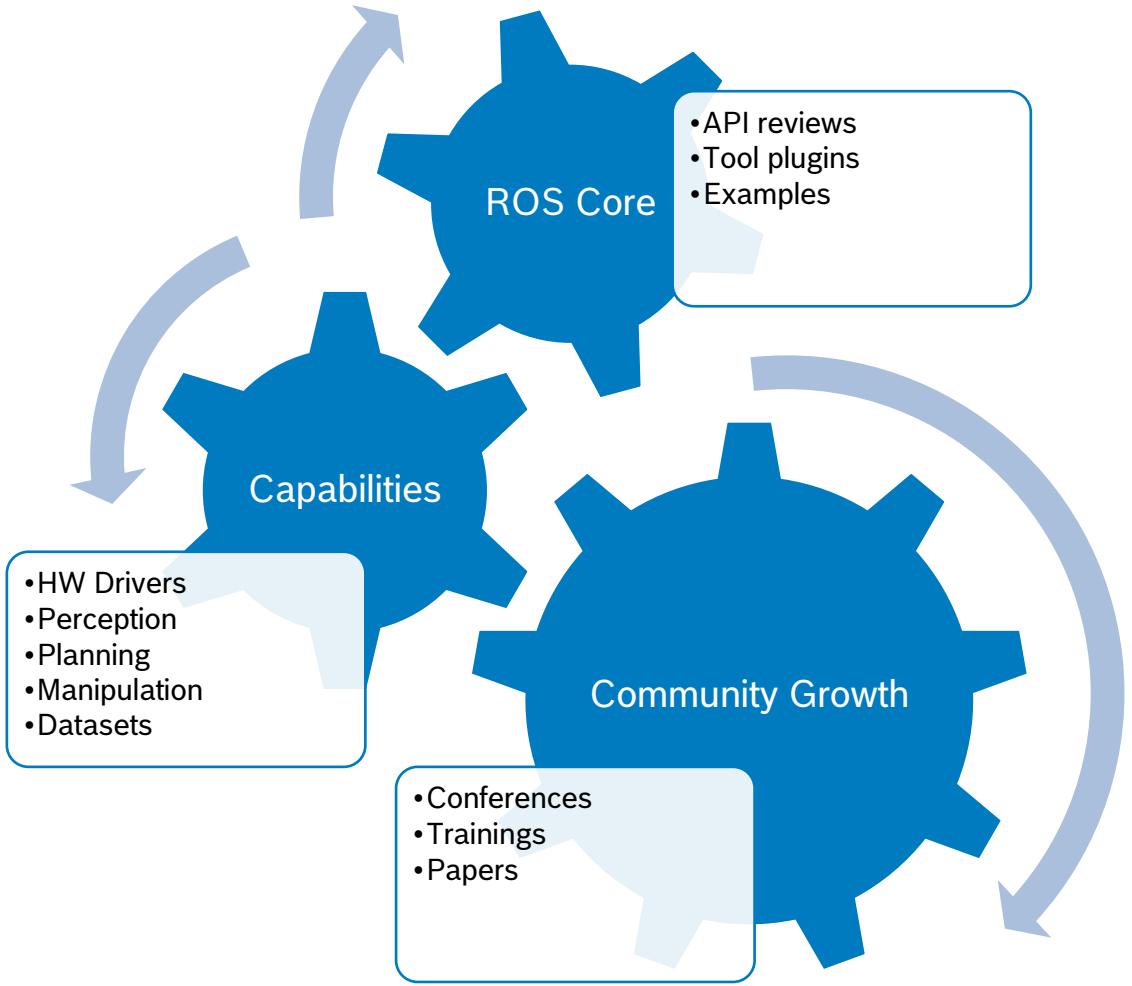
Building an ecosystem







Virtuous cycle



03

ROS in the real world

„The Office Marathon“, 2010

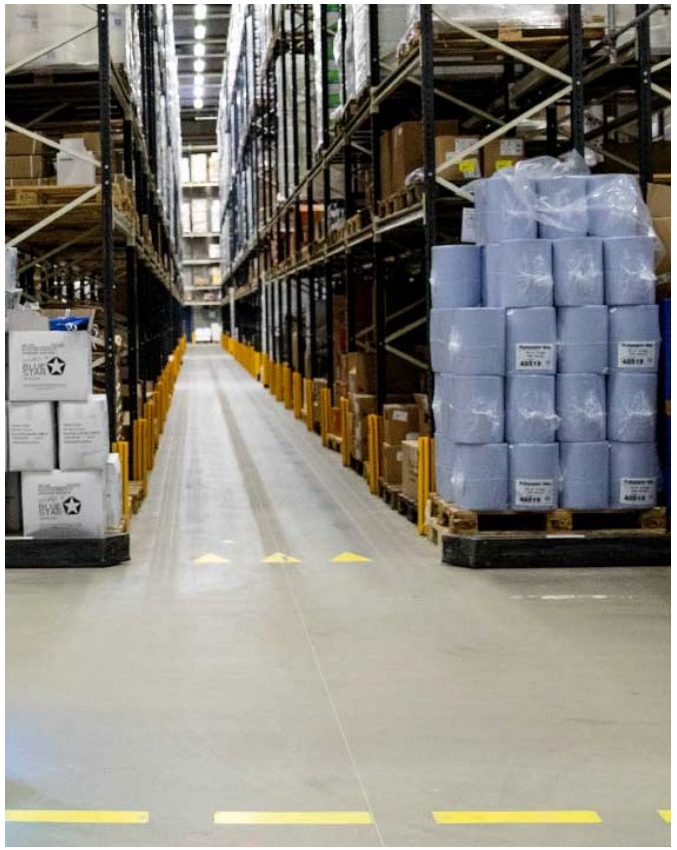
E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The Office Marathon: Robust navigation in an indoor office environment," *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, 2010, pp. 300-307, doi: 10.1109/ROBOT.2010.5509725.

→ 26.2 miles, 30 hours, no collision, one graze.

Can we use this in production?

BOSCH

We tried it in 2015



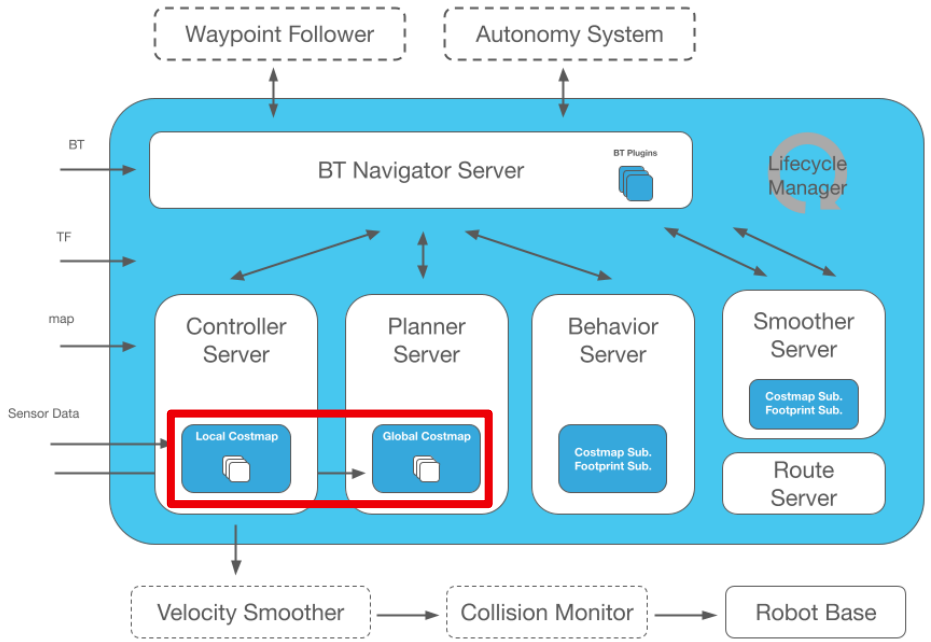
- And we got *really* bad performance in narrow situations

Assumptions vs. Reality

- PR2 in „office marathon“
  - 1kHz odometry
  - 30Hz LIDAR
  - 8 Core Xeon CPU
  - 2.5cm costmap resolution
  - 20 Hz control rate
- Our robot
  - 50Hz odometry
  - 12.5 Hz Safety LIDAR
  - 2 core Atom CPU
  - 10 cm costmap resolution
  - 10 Hz control rate
- Consequences
  - Pose uncertainty larger
  - Inflation radius needs to larger
  - less space to manouver
- Slower speed
- Delayed reactions

On re-use, check assumptions…

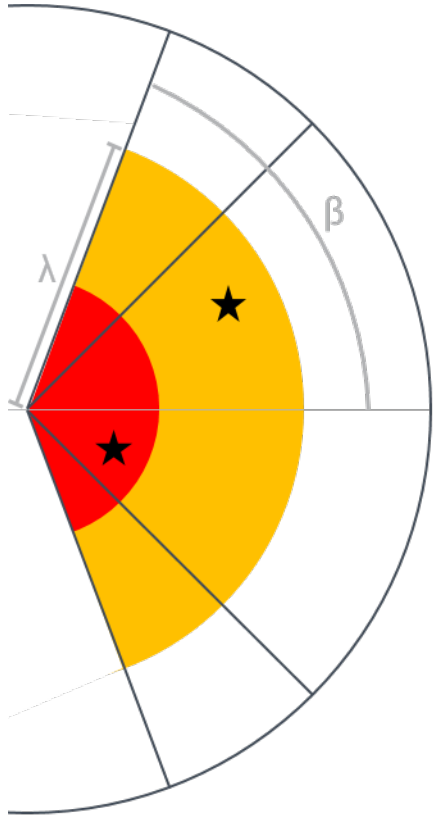
But: A clash of worlds



LIDAR-based safety:

Speed limit zone

Stop zone



The challenge of determinism

Image Source:
K. Adam, A. Butting, R. Heim, O. Kautz, J. Pfeiffer, B. Rumpe, A. Wortmann
Modeling Robotics Tasks for Better Separation of Concerns, Platform-Independence, and Reuse.
In: Shaker Verlag, ISBN 978-3-8440-5319-7. Aachener Informatik-Berichte, Software Engineering, Band 28. December 2017.

- Navigation amongst humans
  - Using ROS move_base
- Problem: Robot kept grazing obstacles that were still at a distance in the costmap

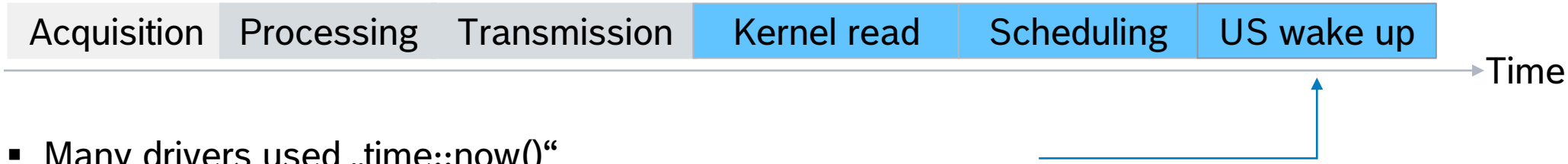
System checks for navigation determinism

- Mismatch between reality and model?
  - → Check sensor rate
  - → Check sensor timing
  - → Check model updates

- Mismatch between control and execution
  - → Check plant model
  - → Check control dead time

Most common sensor timestamping issues

- Basic sensor processing pipeline:
  - Gray is sensor-side, blue is Linux-side



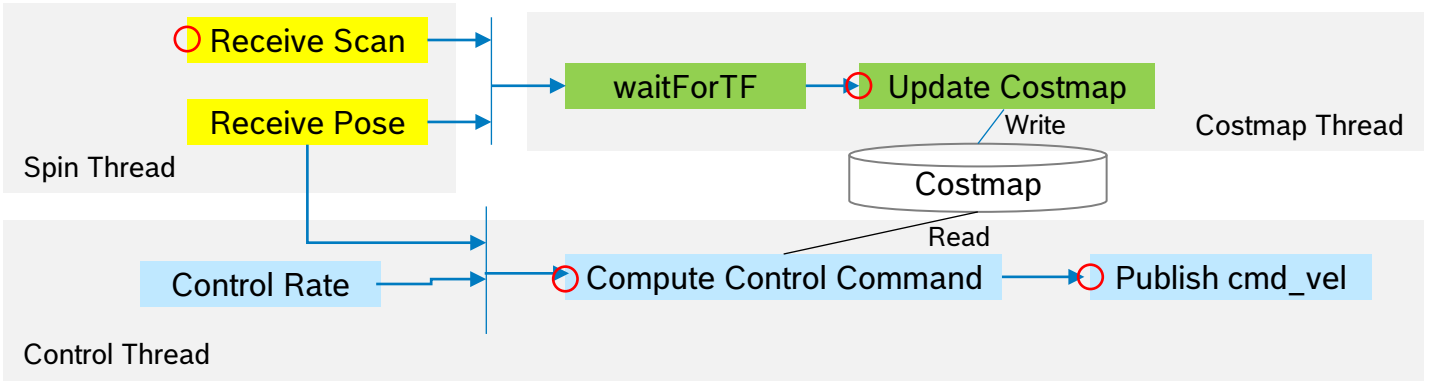
- Many drivers used „time::now()“
  - Too late!
  - → Error up to sensor period (tens to hundreds of milliseconds)
- Some subtract offset for „transmission delay“ or „acquisition delay“
  - But… no real-time scheduling
  - And no priorities in ROS
  - →Error easily 10ms or more due to process scheduling

Still trouble with people

- Dynamic obstacles, such as persons, still pose issues
  - Robot started avoidance motion too late
- We examined sensor data age and found this:

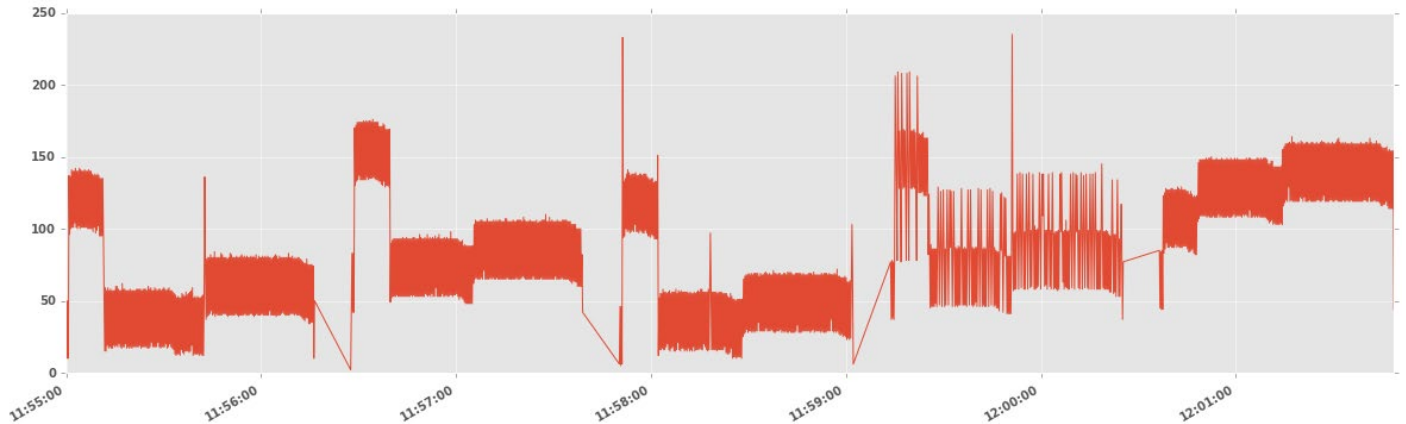
Last-update age of costmap at time of control computation

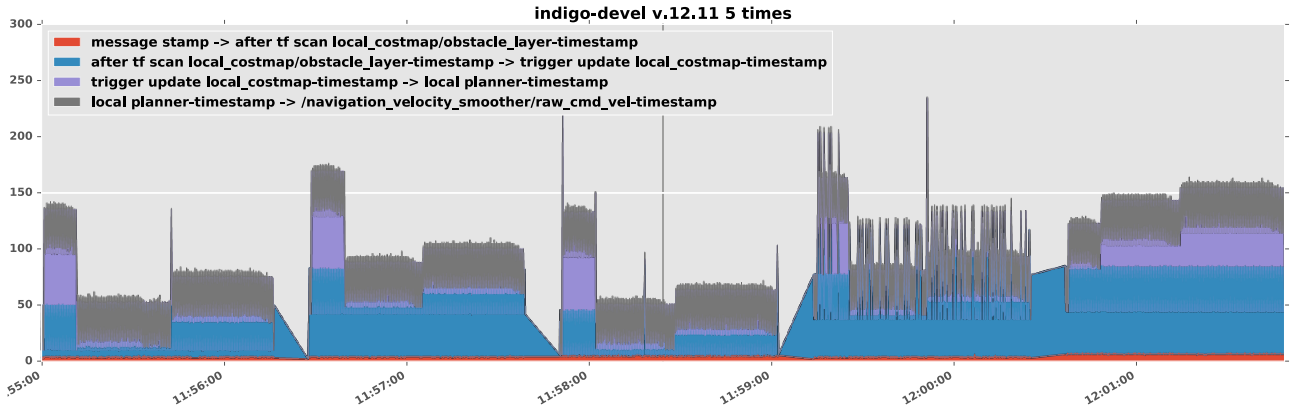
Processing in Costmap



- Time Definitions:
  - T_S^n LIDAR scan n is complete
  - T_U^n costmap is updated with scan n
  - T_C^k local planner starts computing control k
- „Sensor Data Age“: $T_A^k = T_S^m - T_C^k,\ m = \max_{m=1..N}\left(\{T_U^m < T_C^k\}\right)$

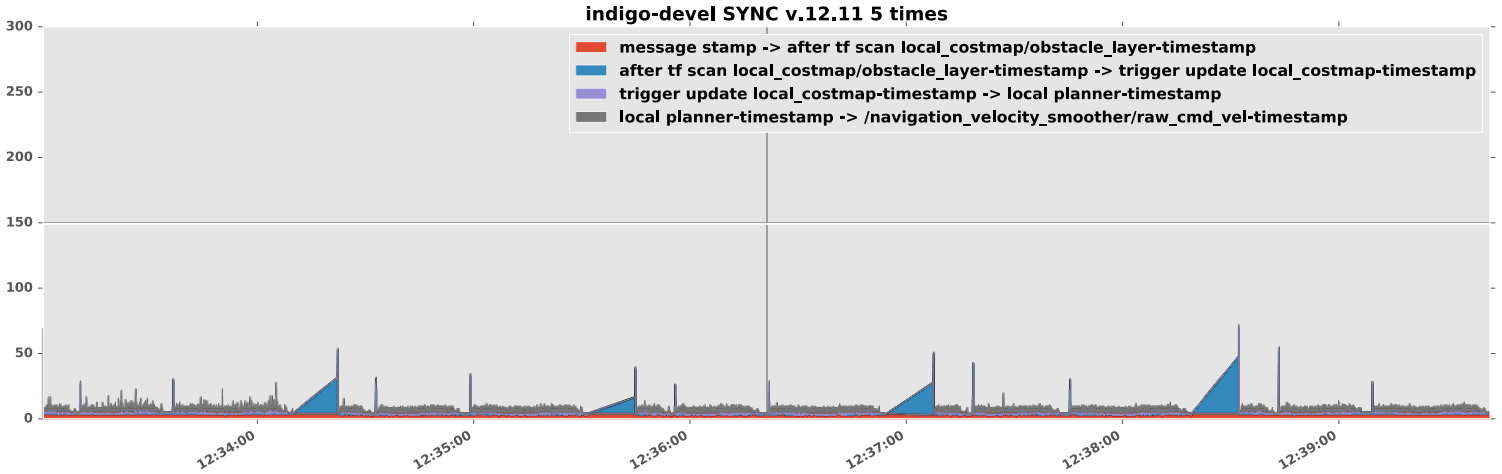
Sensor Data Age Plot





Add synchronization

- Add notification so that controller runs after costmap update
- See ROSCon 2017 talk „Determinism in ROS – or when things break sometimes and how to fix it“ for more details
  - https://vimeo.com/236186712
  - https://roscon.ros.org/2017/presentations/ROSCon%202017%20Determinism%20in%20ROS.pdf



04

Micro-ROS

Micro-ROS

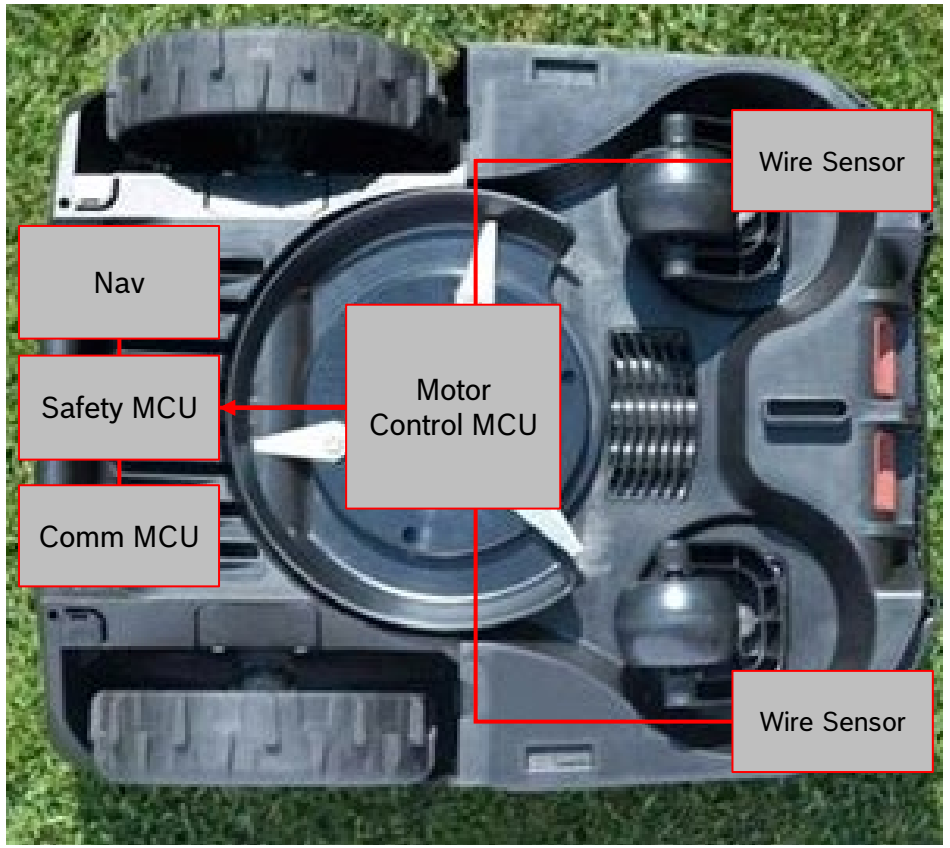
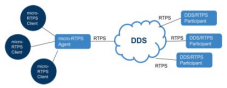
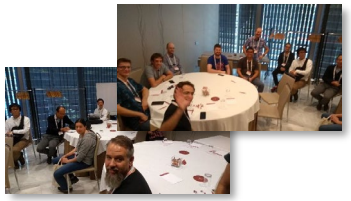


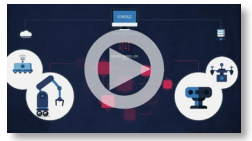
Image source: Bosch PowerTools GmbH, All rights reserved

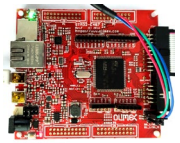
- Microcontrollers have been separate
- Goals
  - Seamless interoperability with ROS 2
  - ROS concepts on the MCU
  - Wide hardware/RTOS support
- And: Better determinism…

Development of the micro-ROS project









2017

- Ideation
- OFERA EU project

2018

- Architecture
- One board
- ROS Embedded WG

2019

- Open-source demo with Turtlebot
- Build tooling for ROS developers
- Start of rclc

2020

- More boards
- Triggered executor

2021

- Build tooling for typical embedded developers

2022

- Successful final review of EU project OFERA

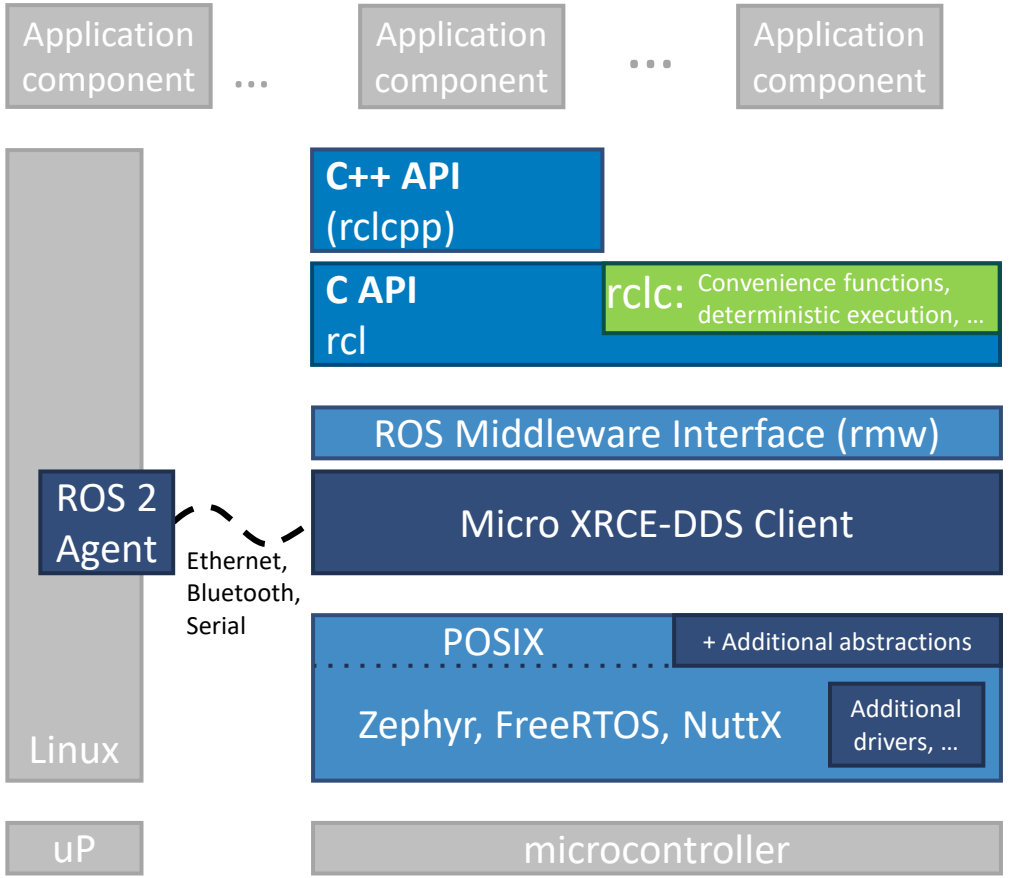
eProsima The Middleware Experts

PIAP Przemysłowy Instytut Automatyki i Pomiarów

FIWARE Foundation

BOSCH

Micro-ROS Architecture



FULL PORTABILITY

Any RTOS and Bare metal Library Generator!

Any low-mid range MCU!

Typical features:

~ 150 KB of flash memory

> 25 KB of RAM memory

General purpose input/output pins

Peripherals: GPIO, USB, Ethernet, SPI, UART, I2C, CAN, etc

REFERENCE HW

- Renesas RA6M5
- Arduino
- Raspberry Pi Pico
- Arduino Nano RP2040 Connect
- ESP-IDF v4.3 & ESP32-S2/C3
- Teensy 3.2 / 3.5 / 4.1 / 4.2
- OpenCR support
- STM32
- Crazyflie 2.1 drone, ...

REFERENCE RTOS

- Mbed RTOS 6.8 / 6.9 / 6.10
- FreeRTOS
- NuttX 10.0 / 10.1
- Zephyr RTOS 2.4 / 2.5
- Azure RTOS ThreadX

Check full list of supported HW & RTOS
https://micro.ros.org/docs/overview/hardware/

Build Systems

For your ROS tooling:

```
$ git clone github.com/micro-ROS/micro_ros_setup.git
$ colcon build

$ ros2 run micro_ros_setup
    ↳  create_firmware_ws.sh [RTOS] [MCU]
    ↳  configure_firmware.sh [MY_APP] -t [TRANSPORT]
    ↳  build_firmware.sh
    ↳  flash_firmware.sh

$ ros2 run micro_ros_setup
    ↳  create_agent_ws.sh
    ↳  build_agent.sh
    ↳  micro_ros_agent [TRANSPORT]
```

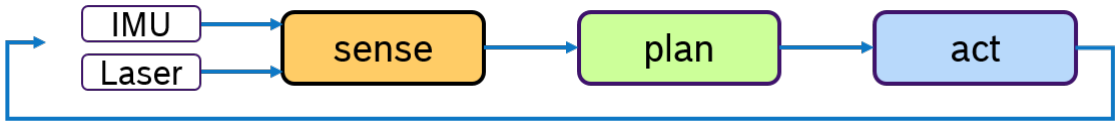
For your favorite tool:

- Renesas e2 studio
- STM32CubeMX/IDE
- RasPi Pico SDK
- Arduino
- ESP32 IDF
- Zephyr

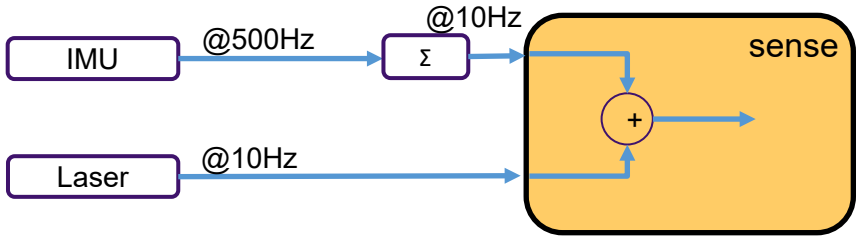
Triggered Executor Concept

Extended API for typical patterns

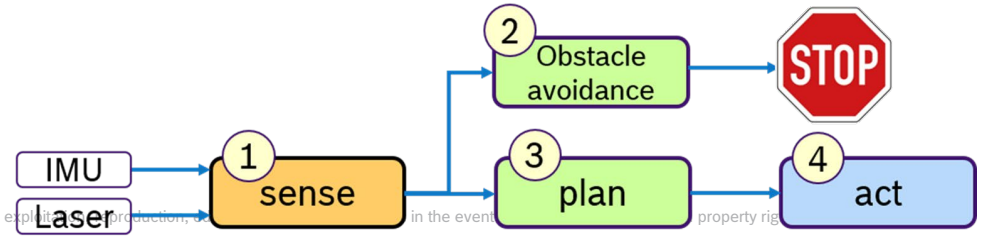
- Control loops



- Data fusion



- Prioritized paths



05

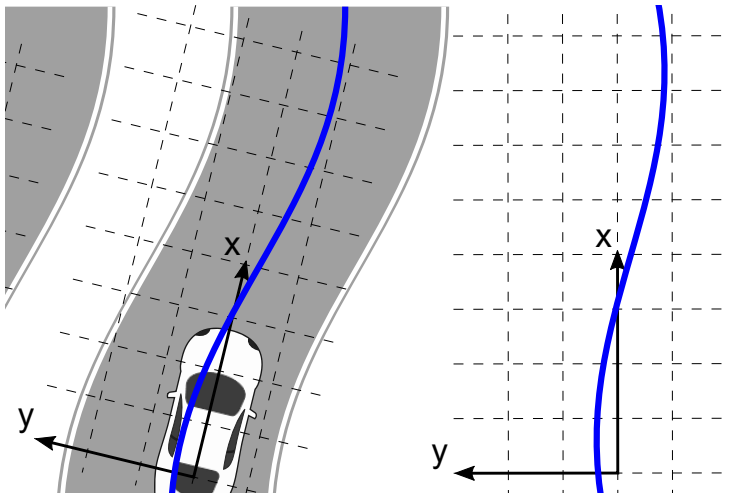
Notes from BR2

BR2 Challenges

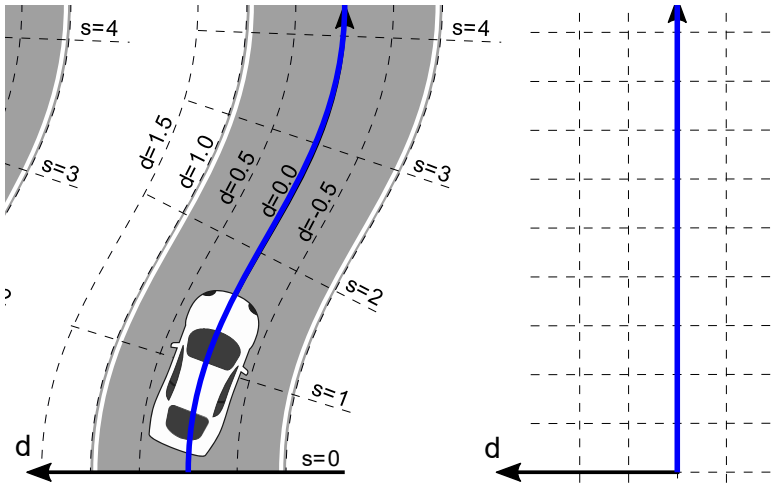
- Rich sensor-set
  - LIDAR, stereo camera rig, environmental sensors, close distance sensors
- Heterogenous compute
  - ARM-based CPUs, GPU, NN-accelerator
- Scalable Autonomy
  - Full autonomy to full tele-op from afar

Leveraging Nav2

- Used for P2P and path-following
- Path-following uses DWB with custom critics to achieve sufficient precision



Cartesian coordinates

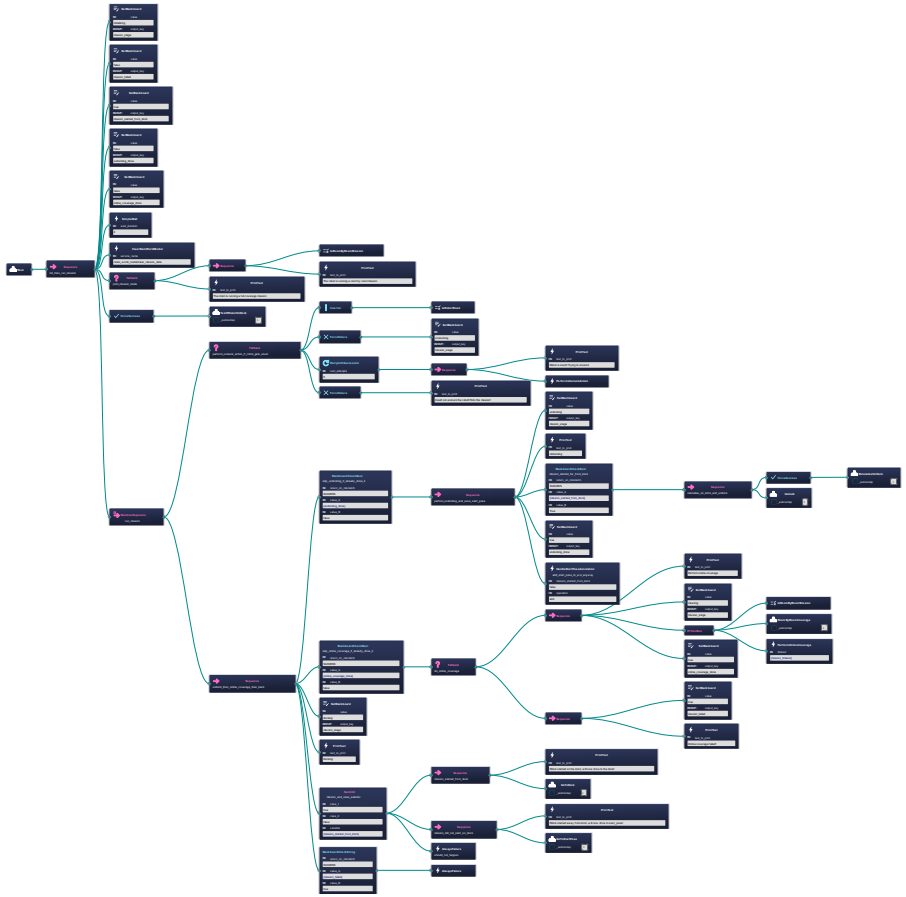


Frenet coordinates

Image source: https://github.com/fjp/frenet

Hierarchical Behavior Tree Approach

- Bosch uses BTs at least since 2015
- Reduce complexity / improve performance through hierarchical BTs coupled with actions
- Image on right is full top-level BT – not so complex, isn't it?
- Pro: Individual trees compact
- Con: Overall structure harder to see

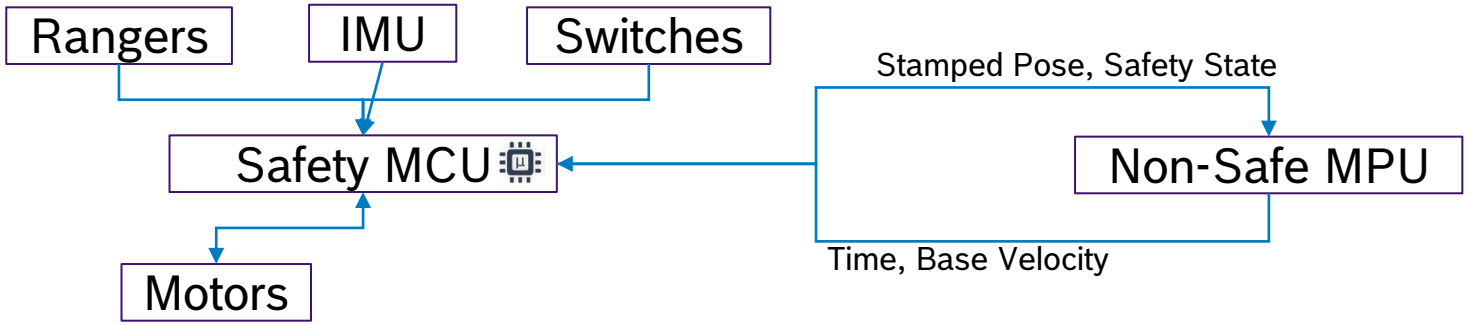


CONVINCE

https://convince-project.eu/

BOSCH

Leveraging micro-ROS

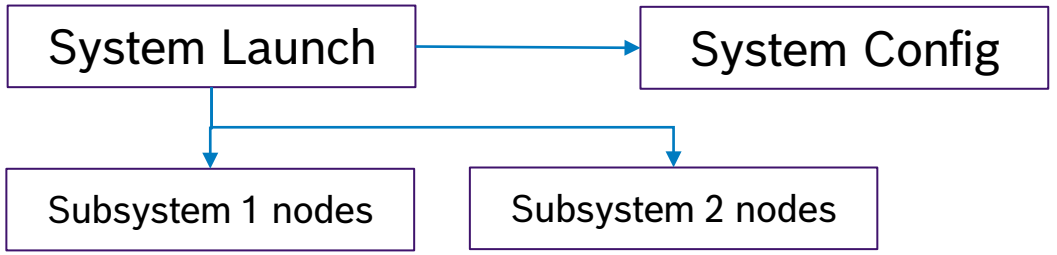


- Pro: Auto-generated messages, time-sync, transport-agnostic
- Con: RAM usage, perceived complexity

2-layer Launch

- Many ROS systems use a deep launch-file inclusion hierarchy
  - This causes a great deal of duplication for passing arguments
- Much content in launch files is also for handling arguments
  - Unecessary complexity

- 2-layer Launch
  - Bottom layer: Nodes for one subsystem
  - Top layer: Only includes from bottom layer
  - All arguments in single YAML file



DDS implementation assumptions vs Robotics needs

DDS Assumptions

- You need different QoS settings
- Throughput is more important than latency
- Network situations are stable
- Users able to configure networking precisely
- Multicast works
- Everybody may want to talk to everybody
- Capacity for plenty of discovery traffic
- One socket is enough for all topics
- Tooling needs to be DDS specific and detailed

Robotics Needs

- Topics need different QoS
- Throughput and latency are both important
- Networks are ad hoc
- Users (generally) know little about networks
- Multicast causes networks to fail
- Most connections are 1-1
- Network capacity is very limited
- Participants need to isolated
- Tooling needs to be integrated

Conclusions

The story continues in 2023…

- Bosch Engineering Group worked with Hako on the autonomous Scrubmaster B75, based on ROS 1

- Meanwhile, Bosch Rexroth also offers the VDA5050 compliant „ROKIT Navigator“, based on ROS 2 Navigation 2

Conclusions

- ROS and ROS 2 have been an important foundation for our research for over a decade
- Journey
  - Tooling for development only
  - Prototyping in early stages
  - Product fully based on ROS 2

- ROS 2 has become more versatile but also more complex
- The layered approach has enabled things such as Micro-ROS and different executors

General

- Know and check your assumptions!
- Become a contributor, it’s worth it

Thank you!

Questions?

(Japanese is okay, will be translated for me)