NODE

CI/CD Tools for Scaling of AMR Software Rollouts

NODE Robotics GmbH

November, 2023

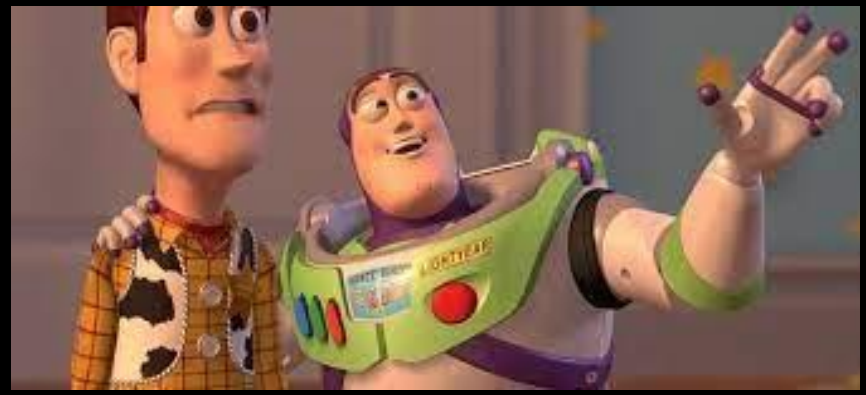
We make mobile robots easy to use.

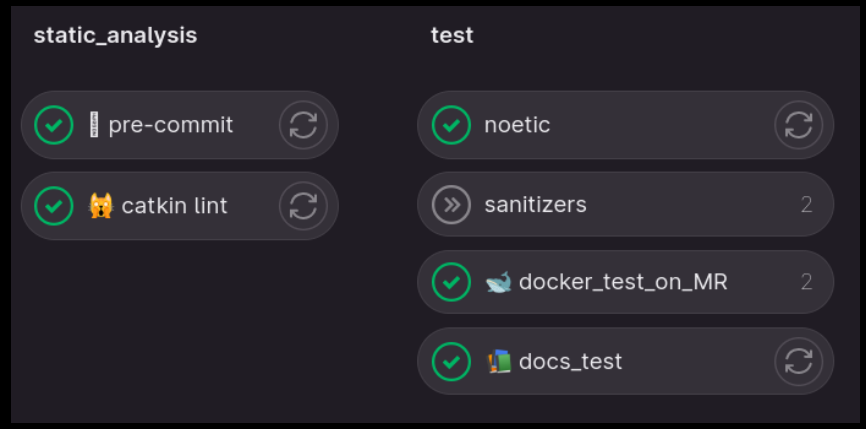
Fraunhofer IPA spin-off, established in 2020 and headquartered in **Stuttgart**, boasts a dynamic team of **30 dedicated employees**. Our remarkable journey has already seen us deploy our cutting-edge software to power **more than 750 Autonomous Mobile Robots (AMRs)**, revolutionizing various industries with their productive applications.

intel ignite · SynerLeap Powered by ABB · VDMA · eXIST · NVIDIA INCEPTION PROGRAM · Fraunhofer · CyberValley · ARENA2036

Situation

- ✓ You've built your awesome Robotics software stack
  - ✓ running on 1000s of robots
  - ✓ everything is going great
- ✓ Edge cases where the behavior can be better
  - ✓ implement some improvement
  - ✓ run unit tests
  - ✓ test in simulation
- ✓ All looks good
  - ✓ click merge and ship?





NODE

Real World

? Are you sure you won't be breaking production on the customer sites?

? Will this improve the experience of all your customers?

? Will this work for all parameter sets that are deployed?

NODE

Multidimensional Matrix From Hell

Robot types

Release v1.0

Release v2.0

Modules

Localization

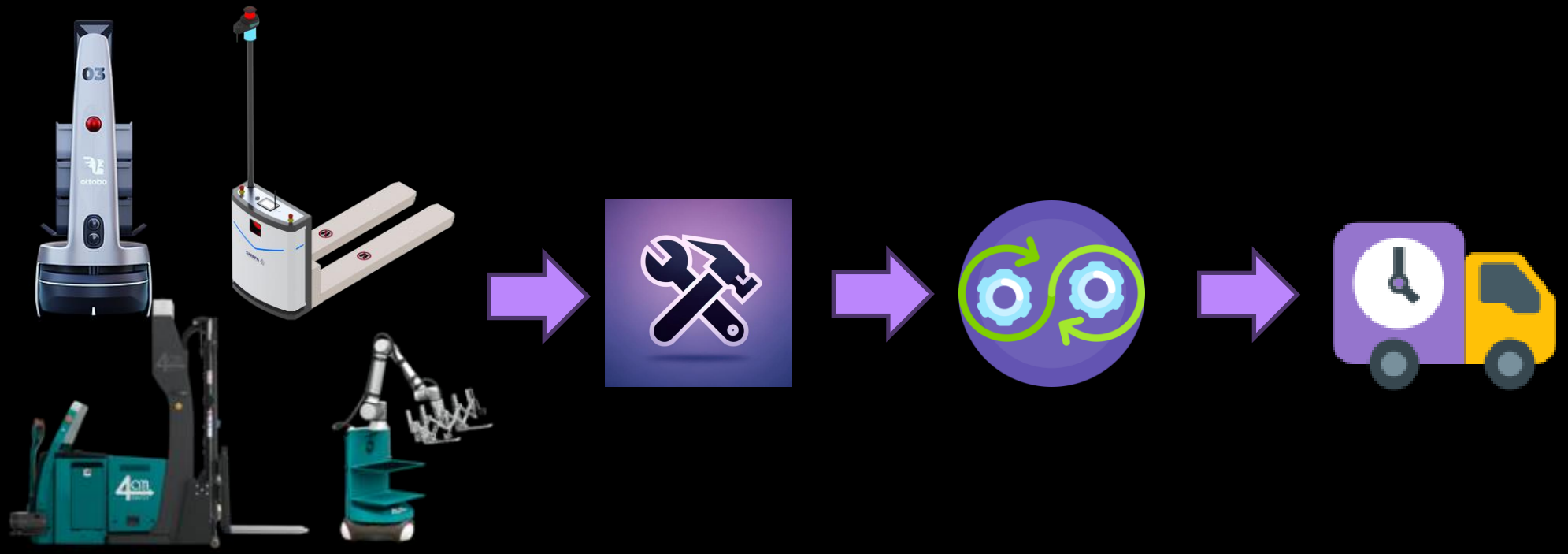
Navigation

Fleet Management

Environments

NODE

What do we want to achieve?



Robots Tools CI/CD Safe Delivery

NODE

How to tackle this problem?

- ✓ Automated Checks
  - check configs for consistency
  - check for missing parameters
- ✓ Automated Testing
  - simulation
  - functional and multi robot tests
- ✓ Real World Data
  - Bagfiles are key
  - automated bag tests

NODE

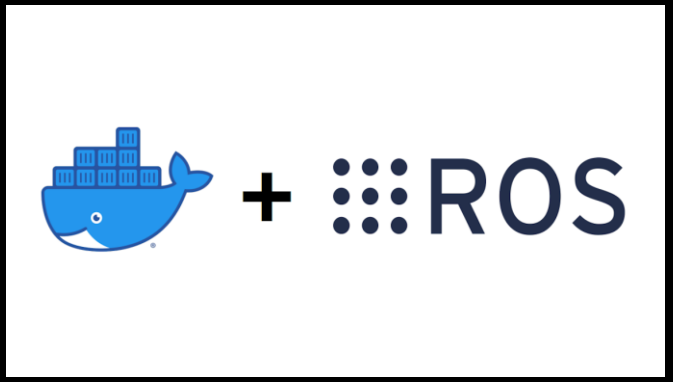
What are the requirements to the tools?

- ✓ Flexible
  - works on a variety of customer setups
  - independent from the hardware
- ✓ Extensible
  - easy to add more component tests in the future
- ✓ Scalable
  - with the # of customers
- ✓ Repeatable and Reproducible
  - reduce the amount of manual work

NODE

How do we deploy software?

- ✓ Robot integrators have a wide variety of setups.
  - different ROS installations
  - different launch structures
  - different architectures
- ✓ Docker based releases
  - reproducible
  - repeatable
  - easier dependency management
  - ROS1, ROS2 compatibility



no more

“works on my machine”

The right setup and tools

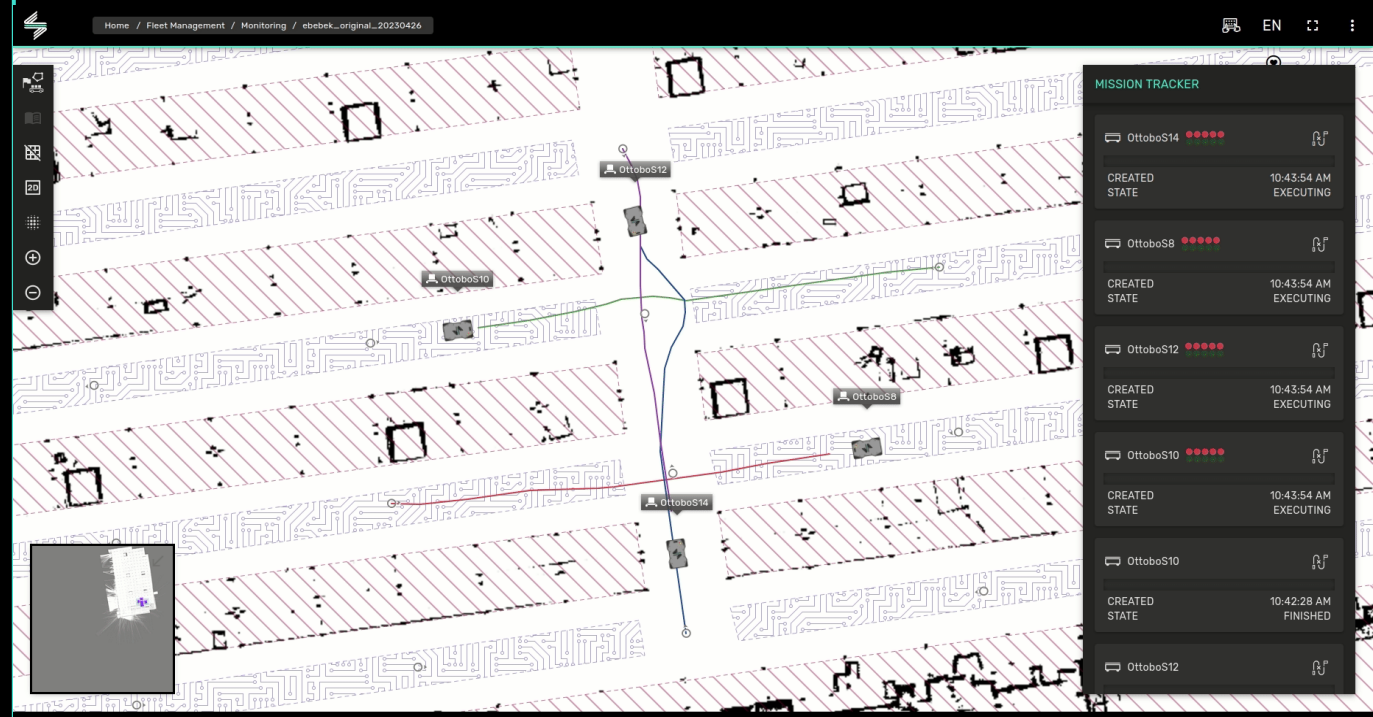
- ✓ docker-compose
  - writing and maintaining them was not easy
- ✓ Python tooling
  - autogenerate docker-compose files
  - spin up containers and simulation
  - check config files
  - API control functionality
- ✓ Light weight simulation and bagfiles
  - laser scan generation
  - individual map support

```
node-tool
Usage: node-tool [OPTIONS] COMMAND [ARGS]...

Options:
    --version                     Show the version and exit.
    -r, --release-file FILENAME   The release file to use
    --latest                      Use the latest release instead of providing a file
    --latest-stable               Use the latest release candidate from the global release repo
    -c, --robot-config FILENAME   [DEPRECATED] The robot config to load
    --help                        Show this message and exit.

Commands:
    config-checker  Run config checker tool
    config-specs    Dumps the configuration specification for a given...
    dump-docs       Dumps all API specs from all components for the current...
    dump-release    Dumps the release with all components filled in from...
    env-server      Environment server tooling
    fleet-server    Fleet server tooling
    generate        Generate docker-compose files
    get-image       Prints the image that is used for this component
    simulate        Generate and run simulation
```

NODE

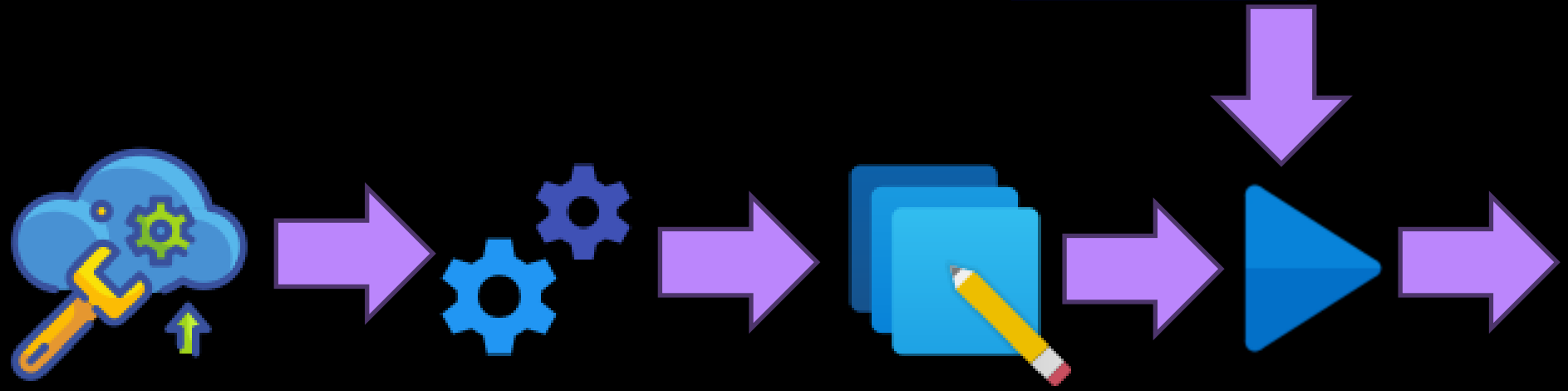
The right setup and tools

- ✓ Navigation tests
  - generalized tests for navigation
  - module specific tests for docking
  - API based testing
- ✓ Multi-Robot tests
  - test whole stack with custom setup
- ✓ Customer specific tests
  - test individual behavior and interfaces
  - custom bagfile testing

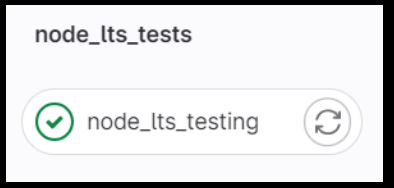


NODE

NODE Tool Pipeline



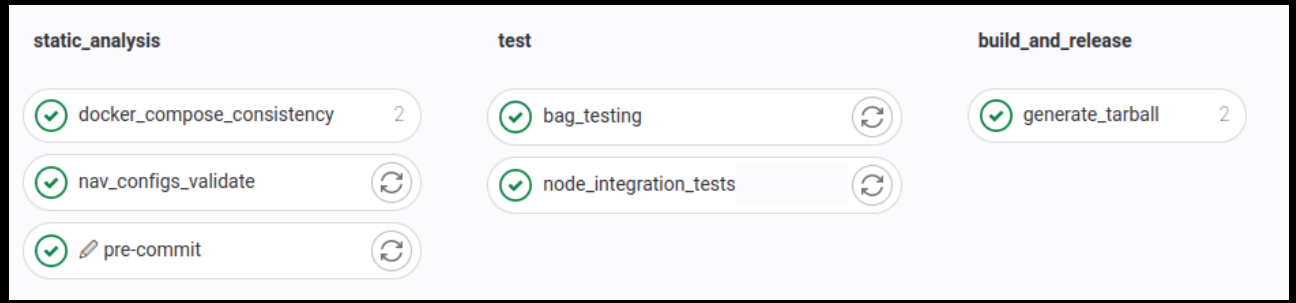
NODE Tool Config set Docker compose Spin up stack



NODE

Implementation in CI

- ✓ CI Templates that can be included across repos
  - include the testing job
  - Inherit from base ci job
  - specify config paths in the repo
  - run any ROS bag-based or simulation test
  - test customer setup directly




```yaml
bag_testing:
  extends: .node_lts_tests
  variables:
    PROVISIONING_RELEASE_FILE: "${CI_PROJECT_DIR}/provisioning/release.yaml"
    PROVISIONING_NAV_CONFIG_FILE: "${CI_PROJECT_DIR}/configs/robot_config.yaml"
    PROVISIONING_BAG_CONFIG_FILE: "${CI_PROJECT_DIR}/configs/bag_test.yaml"
    TOX_OPTIONS: "-e py311"
  tags:
    - BAG_TESTING
```

NODE

Summary and next steps

- ✓ Infrastructure in place
- ✓ Add further testing of new modules
- ✓ Empower tooling even further for new deployments and use cases
- ✓ Build also tooling for deployment of different robots of the same type

NODE

Philipp Schnattinger
Chief Robotics Officer

philipp.schnattinger@node-robotics.com
Phone: +49 17695860092

Get in touch!

www.node-robotics.com