Mobile robotics scale-up leveraging ROS

Guillaume Doisy - Lead System Architect – @doisyg
Marcus Scheunemann – Head of Autonomy - @scheunemann
ROSCon 2024 – 23/10/2024

DEXORY

Context

ROS 2 developer experience at Dexory

Software integration

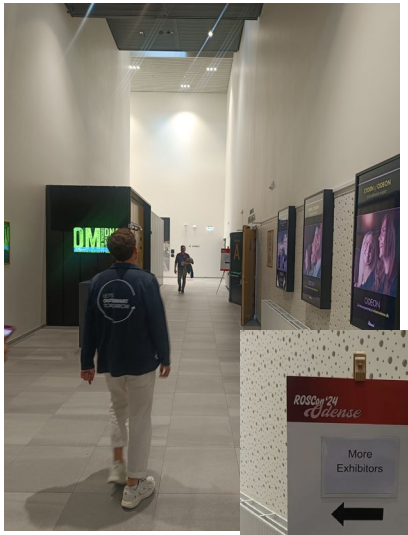
Bug hunting diary

Working with ROS open source software

DEXORY

Context

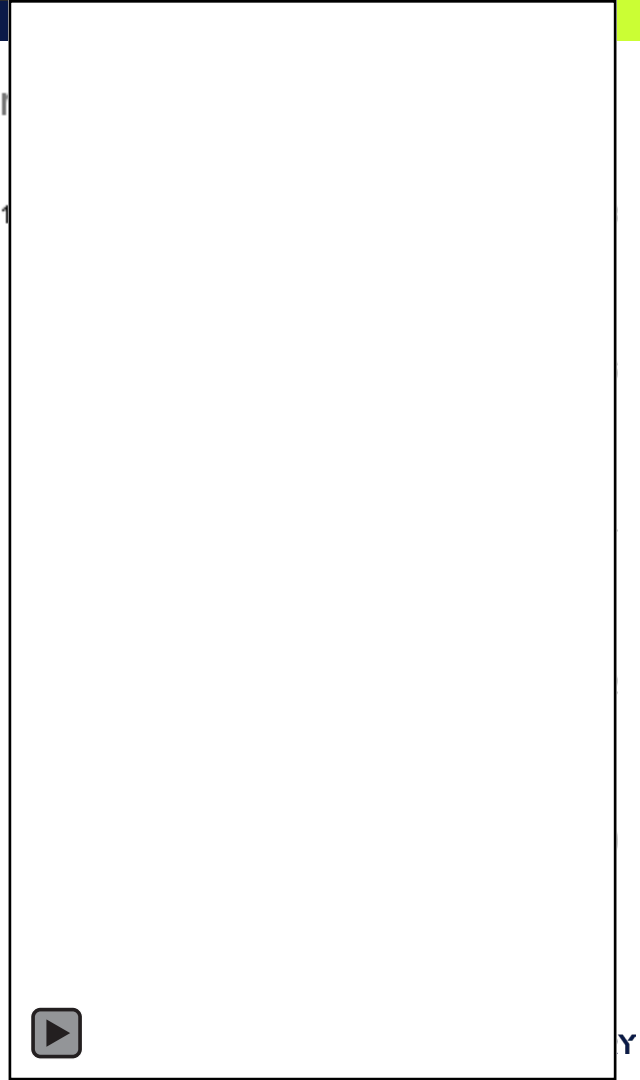
Dexory’s AGV



Context

Dexory's AGV

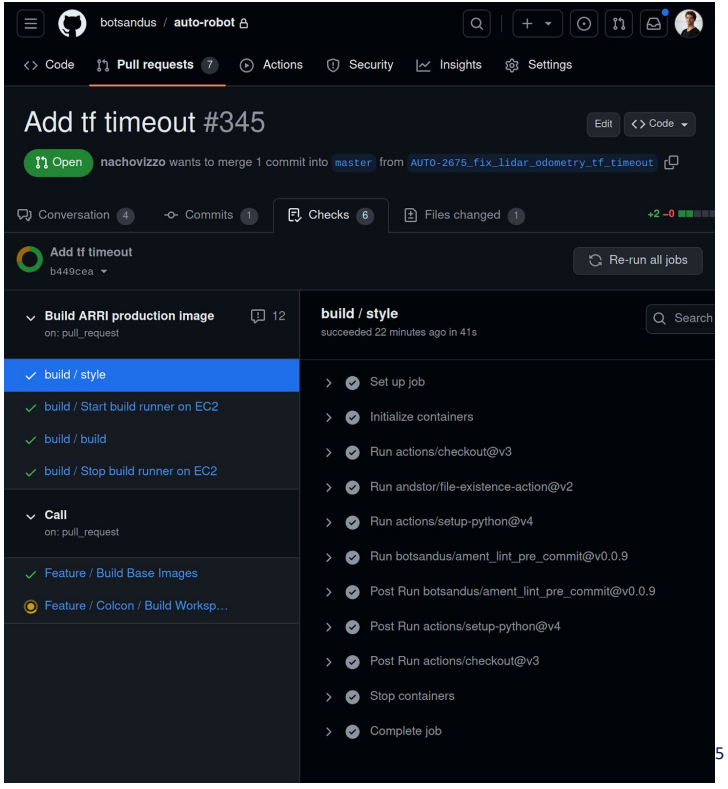
- 600kg / 14 meters scanning autonomous mobile robot
- Up to 185 000 m2 (2M ft2!) warehouse, and counting…
- 1 robot per warehouse
- 6x 3D lidars + 2x 2D lidars
- First CAD in September 2022 – 15 people company
- First deployment in March 2023
- Today (October 2024):
    - 180+ people
    - Strong funding (80M Series B)
    - Over 90 units built in the last 12 months
    - Currently deploying 3 robots a week
    - Global customers:
        - UK
        - Pan EU
        - USA
        - MENA



ROS 2 developer experience at Dexory

DevOps

- Get DevOps resources and CI/CD running early
- Standardized developer environment (docker or other), no more “it runs on my machine”. With some flexibility (e.g. not only VS Code)
- Linting (pre-compilation)
- (fast) CI build check
- (fast) CD binaries generation
- Dependency control
- Allow non-persistent production image alterations (launch files, parameters files, packages)

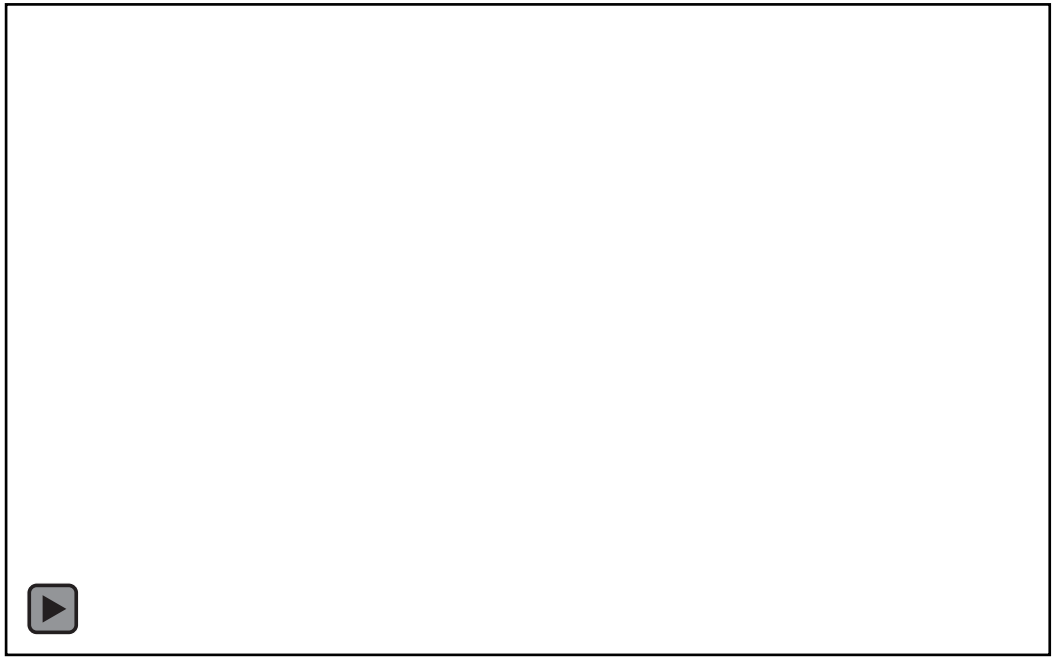


Pro tip: visit “Session 1a - ROS testing and tools” at 13:50 with Ruffin and Marcus.
Accelerating the CI/CD-to-robot cycle by 10x for 1/10th the cost

ROS 2 developer experience at Dexory

Simulation

- All sensors and actuators are worth simulating, even roughly (e.g. charging behavior)
- Test sensor processing with bags
- API pipeline and behaviors with a simulation environment
- RTF >= 1.0 running on isolated developers machines. With 8 lidars thanks to the Gazebo Robotec.ai RGL plugin.
- For mobile robots, 2.5D simulation: topological simulation (i.e. customer site reproduction)



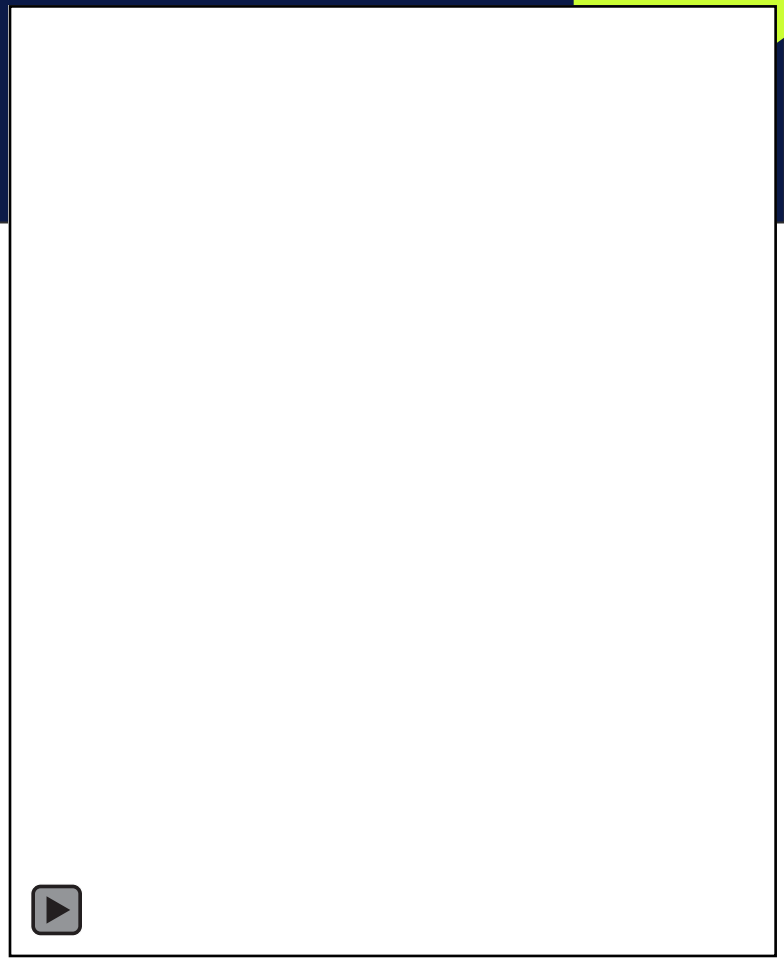
ROBOTEC AI

https://github.com/RobotecAI/RGLGazeboPlugin
https://github.com/RobotecAI/RobotecGPULidar

ROS 2 developer experience at Dexory

Tooling

- Make ssh easy for developers, leverage VPN / Wireguard / Tailscale
- Easy ros bag recording and downloading.
- Record raw, replay with filter and sensor pipeline launch files (SIMULATION / use_sim_time).
- Live visualization of data. Don’t reinvent the wheel: Rviz + VNC
- Logs, logs all the time and everywhere (with Grafana / Telegraf / Prometheus).



ROS 2 developer experience at Dexory

Keep it simple, test continuously test, deploy early

- Developing new features in a RaaS scale-up is (also) **hard**
- specifically when you need to keep existing customers happy

- Challenge:
  - each new feature increases complexity!
  - thus each change increases efforts for later features

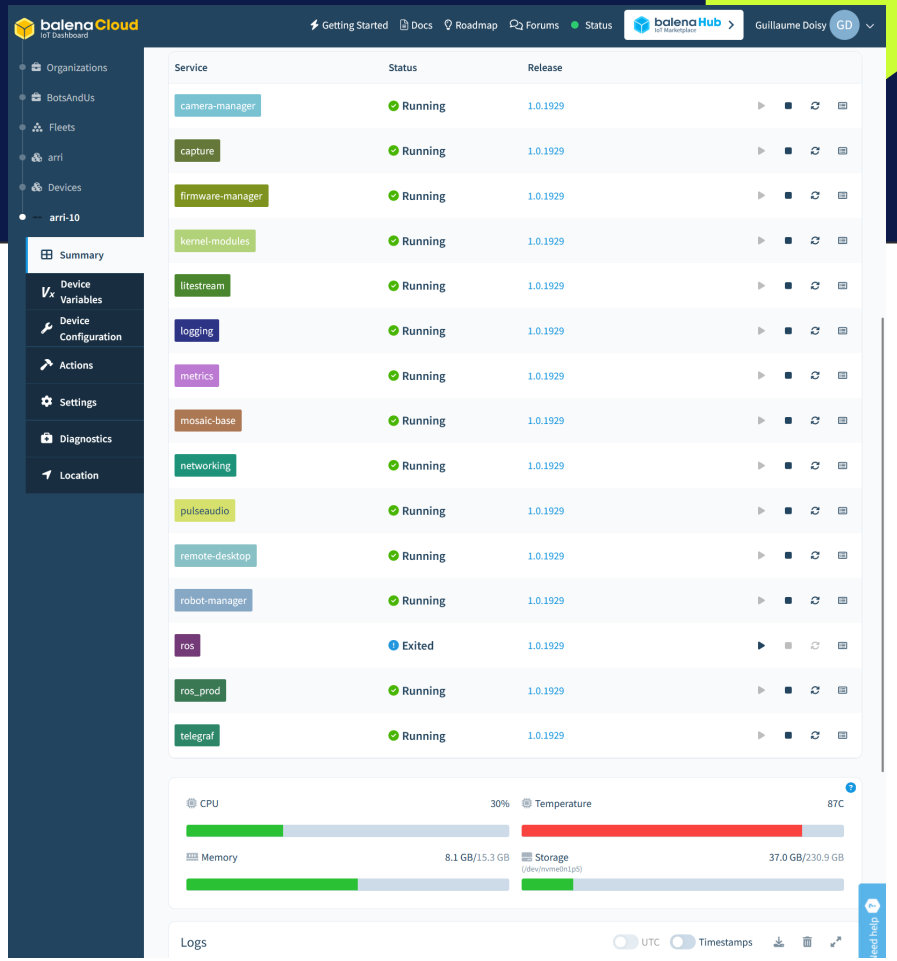
Solution:
  - Keep features minimal, focus only customer needs
  - deploy early

Pro-tip:
- Do not call it testing (it's boring; it feels like nothing is done)
- link it to increasing "annual contract value" of a customer subscription (OKR); coin it as "decrease feature time to market" (KPI)
- Get people on testing (test engineer; QAs)

Software integration

Packaging

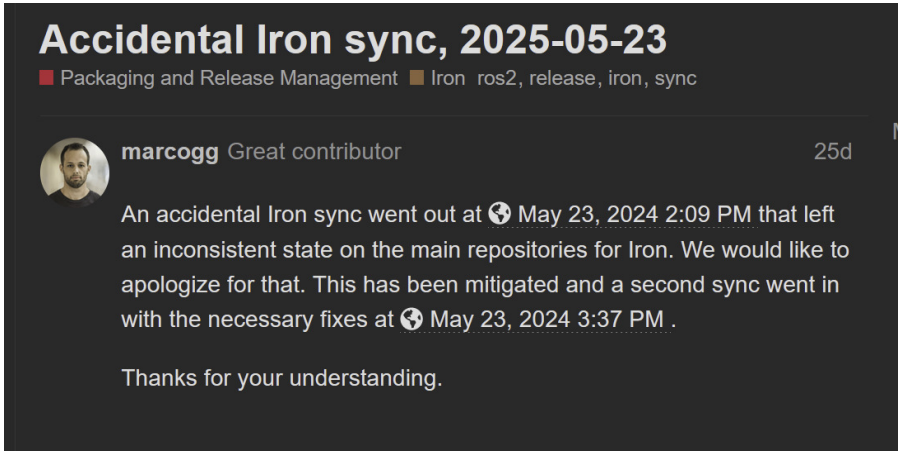
- Tried historically (pre-Dexory):
  - Ubuntu Desktop with building from source (don't do that)
  - Ubuntu Desktop with CI generated debian packages (blooming and versioning hell)
  - Ubuntu Desktop with Docker (but Ubuntu Desktop is not an IoT distrib, boot will be destroyed at some point)
  - Ubuntu Core with snap (cleanest approach so far but… snaps)
  - Yocto
- Today:



Software integration

Controlling external dependencies

- Fix everything: binaries, sdk, sensor driver, etc…
- Why ?
    - Trust open source with control
    - Upstream can disappear
    - Uncontrolled updates and regressions
    - Independency of the build chain
- From source: fork
- From binaries: mirror repos, classically Ubuntu + ROS with aptly



Software integration

Taming DDS

- Biggest ROS 2 entry pain point
- Some complains:
  - Very not optimized default settings for my standard use-case
  - Cross-talk by default
  - One drama per ROS update
- Our Iron recipe:

```
export RMW_IMPLEMENTATION=rmw_cyclonedds_cpp
export ROS_AUTOMATIC_DISCOVERY_RANGE=SUBNET
# Or, if no need for network com:
# export ROS_AUTOMATIC_DISCOVERY_RANGE=LOCALHOST
export CYCLONEDDS_URI=${CYCLONEDDS_URI:-
"<CycloneDDS><Domain><General><Interfaces><NetworkInterface
name=\"lo\"/></Interfaces><AllowMulticast>true</AllowMulticast></Gen
eral><Discovery><ParticipantIndex>none</ParticipantIndex></Discovery
></Domain></CycloneDDS><Gen><Allow>spdp</Allow></Gen>"}

ip link set lo multicast on
sysctl -w net.core.rmem_max=2147483647
```

Software integration

Time synchronization

- Choose sensors with PTP support (if you can)
- Use your computer network interface as PTP hardware grandmaster (no need for fancy equipment)
- Synchronize hardware clock with system clock:

```
echo "Starting ptp4l and phc2sys for PTP provision..."
/usr/sbin/ptp4l -f /etc/linuxptp/ptp4l.conf boundary_clock_jbod=1 clockClass=128 -i ${WIRED_IFACE} &
/usr/sbin/phc2sys -w -s CLOCK_REALTIME -c ${WIRED_IFACE} &
```

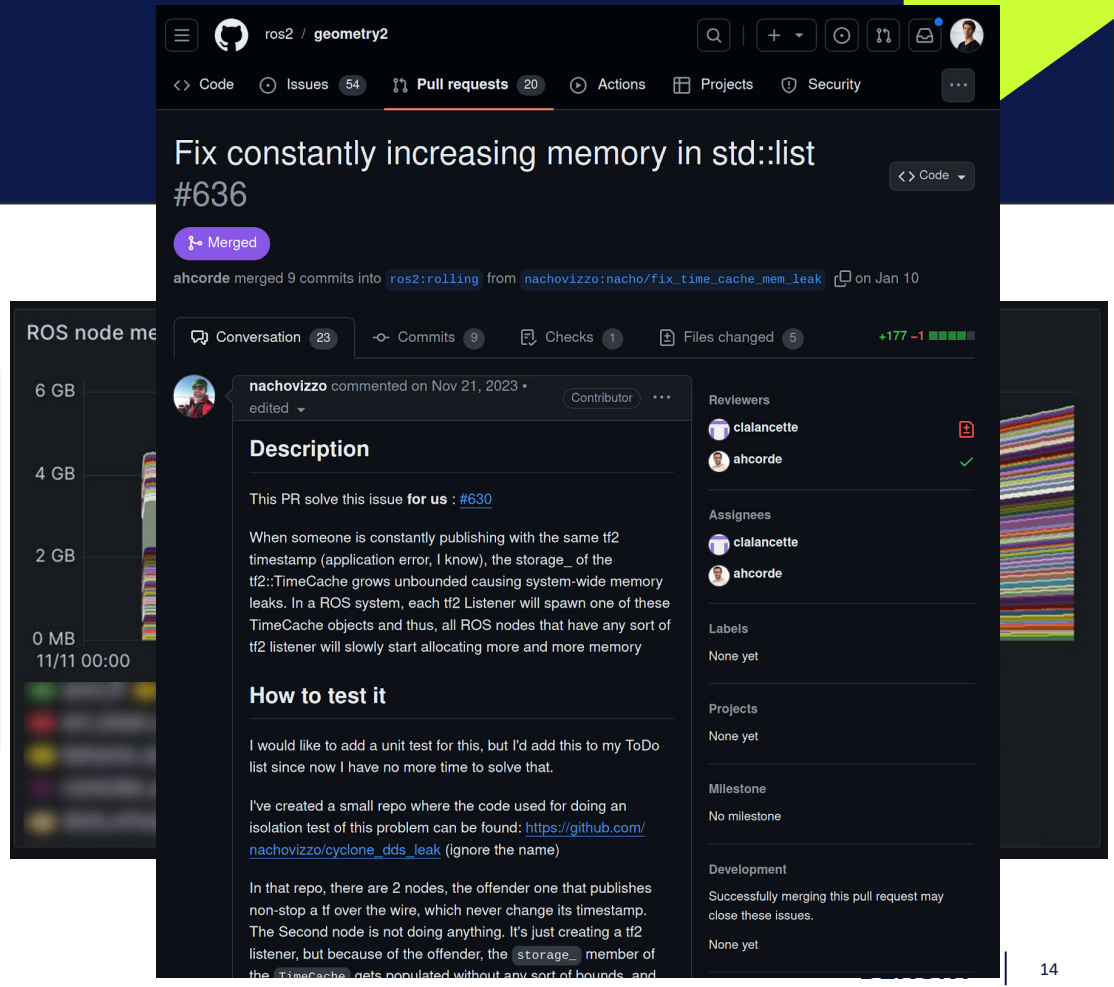
- Synchronize system clock with world time with NTP (chrony)

Bug hunting diary

Bug hunting diary

Elusive memory leak

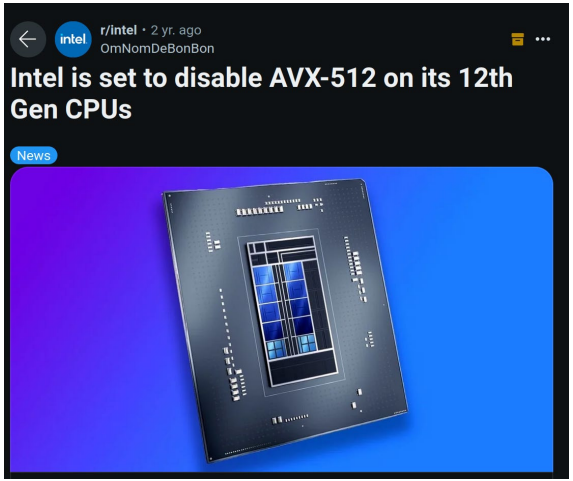
- Only appears after a specific dock sequence
- Hunted with Grafana
- Publishing outdated TF at high frequency is increasing memory usage on all TF listeners
- Fix merged: https://github.com/ros2/geometry2/pull/636 (thanks Alejandro)



Bug hunting diary

Intel instruction set fun

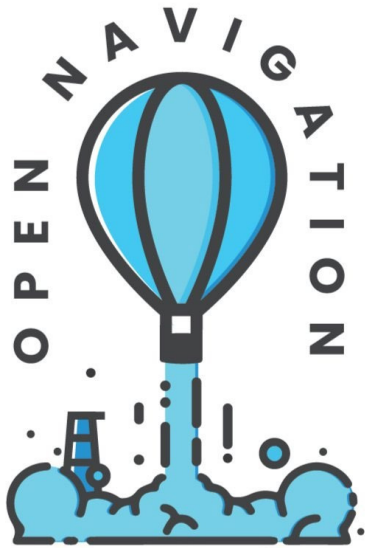
- Binaries built locally work / Binaries built by CI crash at runtime
- Works on NUC11 / Crashes on NUC13
- Similar nav2 story



https://github.com/ros-navigation/navigation2/issues/3767

Working with ROS open source software

- Advantages and drawbacks of using an updated version of ROS 2
- Value of open source collaboration for Dexory
- Feedback of 1.5+ years of sponsoring Open Navigation / nav2
- Concerns about the maintenance of the core ROS packages / OSRA future



open source robotics alliance

Thanks !

(we are recruiting)

Guillaume Doisy - Lead System Architect – @doisyg
Marcus Scheunemann – Head of Autonomy - @scheunemann
ROSCon 2024 – 23/10/2024

DEXORY