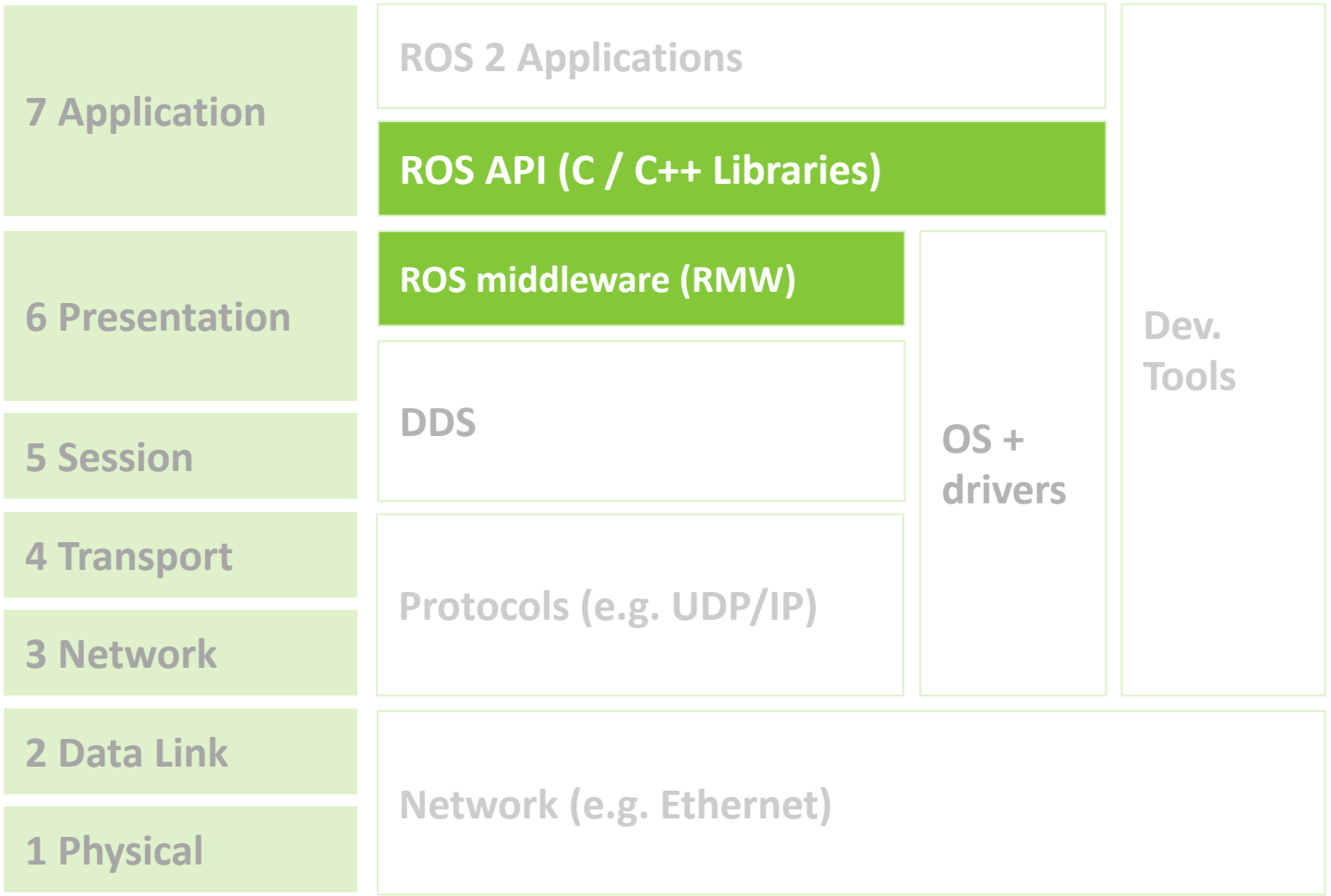
ROS 2 EXECUTION

INGO LÜTKEBOHLE

BASED ON WORK WITH:
TOBIAS BLASS, JAN STASCHULAT, RALPH LANGE AND CHRISTOPHE BOURQUE BEDARD

BOSCH



BOSCH

To achieve (real-time) guarantees, execution must be deterministic.

BOSCH

BACKGROUND

The content in the following slides is accurate for all ROS 2 versions up to, and including, the upcoming Eloquent release.

BOSCH

ROS 2 Execution
Expectations on execution

- Sources of (framework-level) events
  - Timers
  - Messages coming in
- All other things being equal, we expect reactions to events in the order of occurence
- For real-time, a reaction must always occur within a certain time-frame
- Recall ROS 1
  - Network thread inserts messages into queue upon complete receiption
  - Callbacks for a message are executed in registration order
  - For timers, inserted into queue upon firing

→This is, essentially, *best-effort FIFO*

ROS 2 Execution

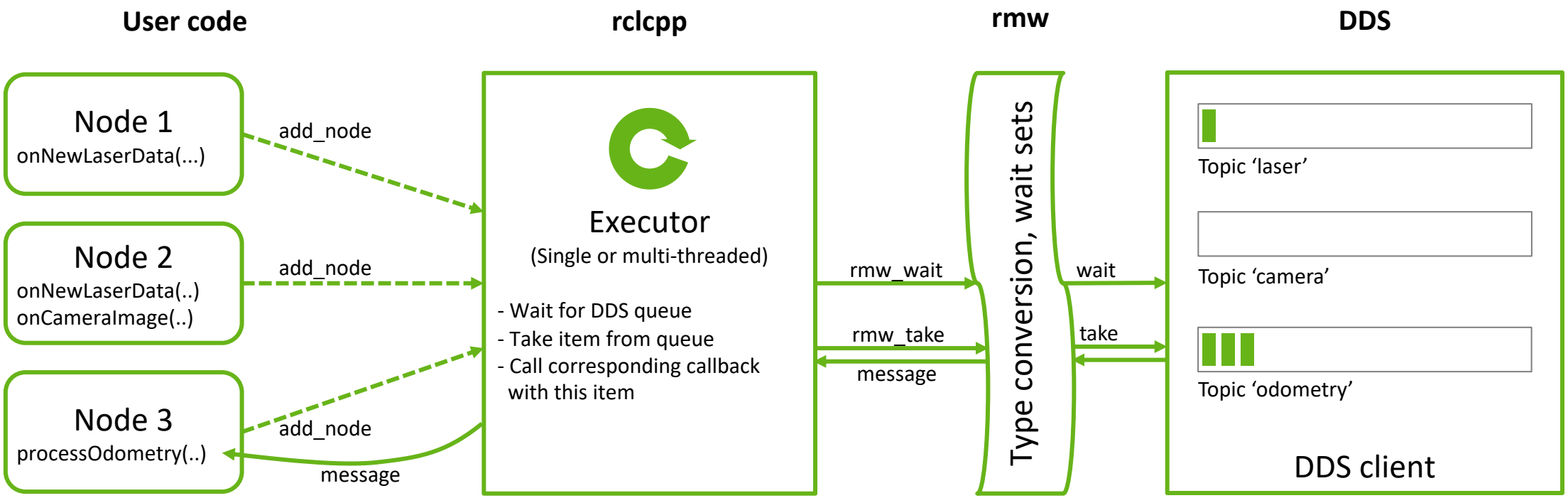
Expectations on execution in ROS 2

- What do we expect ROS 2 to do?
- Without special configuration? → Same as ROS1
- But with more capabilities…
  - Maybe priorities, some topics are more important
  - Maybe guarantees on latencies
  - Maybe use DDS-send-timestamps to reduce network effects…
  - In general, „more deterministic“
  - And, of course, zero-copy, etc. pp.

- Unfortunately, none of the above is provided

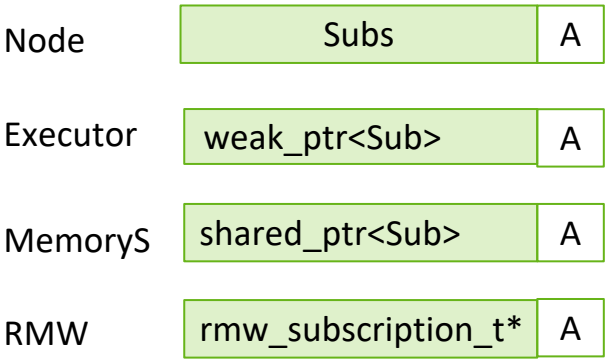
ROS 2 Execution

How ROS 2 places the executor



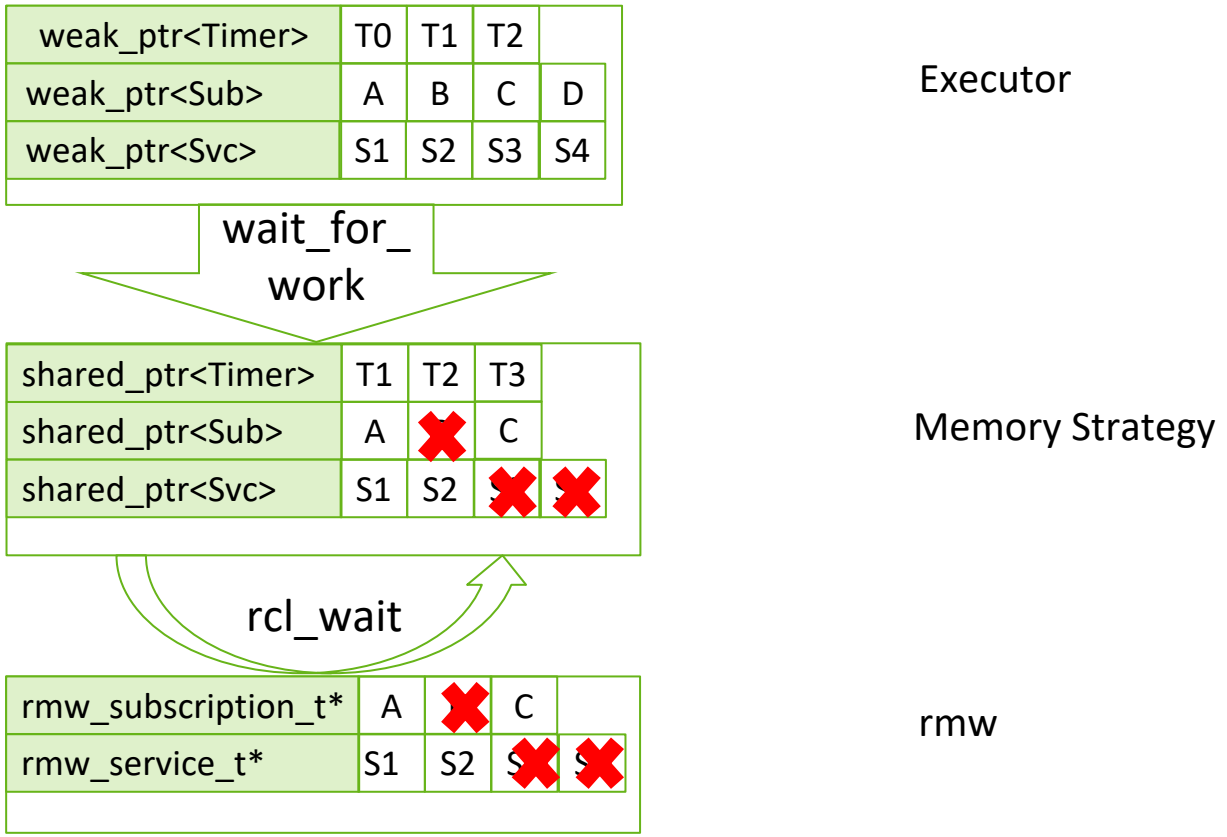
ROS 2 Execution
Part 1a: Collect event status

- rmw_wait is similar to select/poll
  - „Given a list of communication identifiers, tell me which ones are ready to process“
  - Note: Binary information
- Data structures strongly influenced by lifetime considerations
  1. Nodes *contain* timers, subscriptions, services, etc.
  2. Executor holds *weak_ptr*‘s to these
  3. During spin, the so-called „memory strategy“ obtains *shared_ptr*‘s
  4. The RCL and RMW layers only see plain „C“ pointers

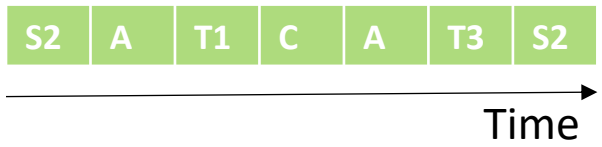


ROS 2 Execution

Part 1b: Information flow through the layers

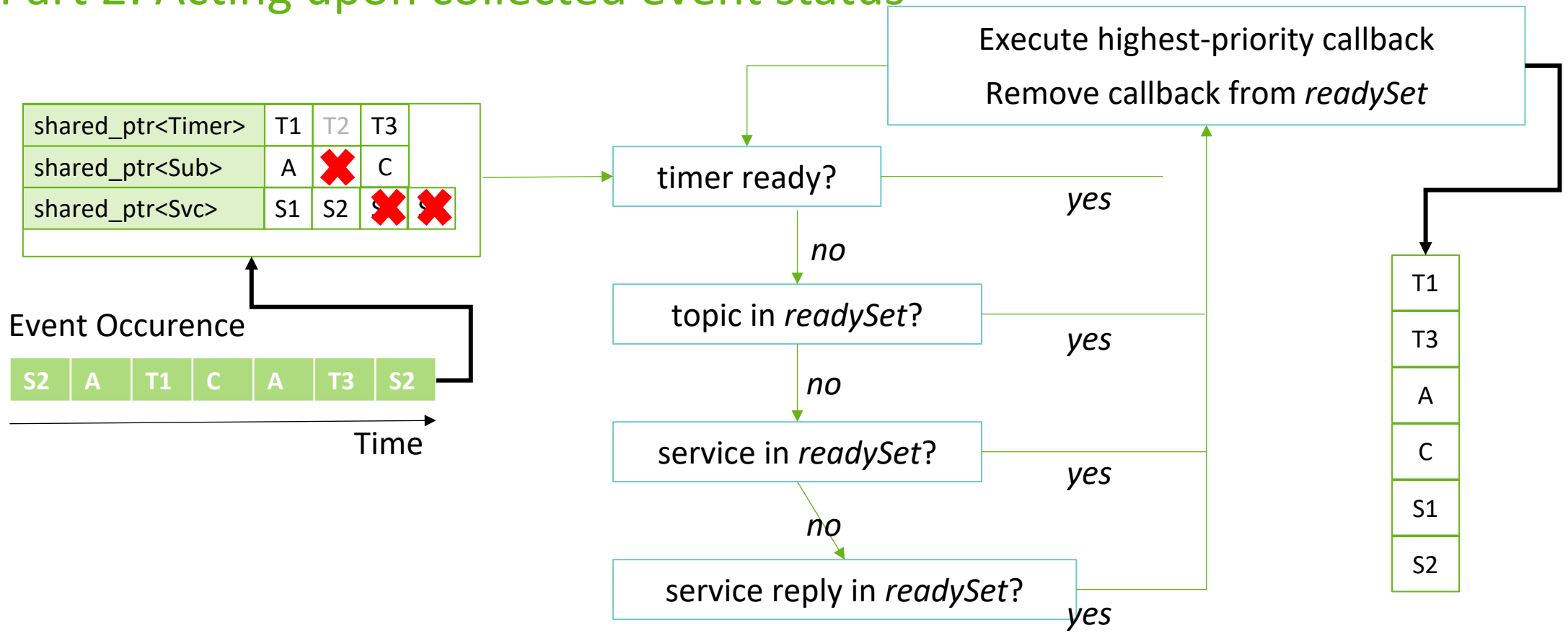


Events



ROS 2 Execution

Part 2: Acting upon collected event status



ROS 2 Execution: Determinism

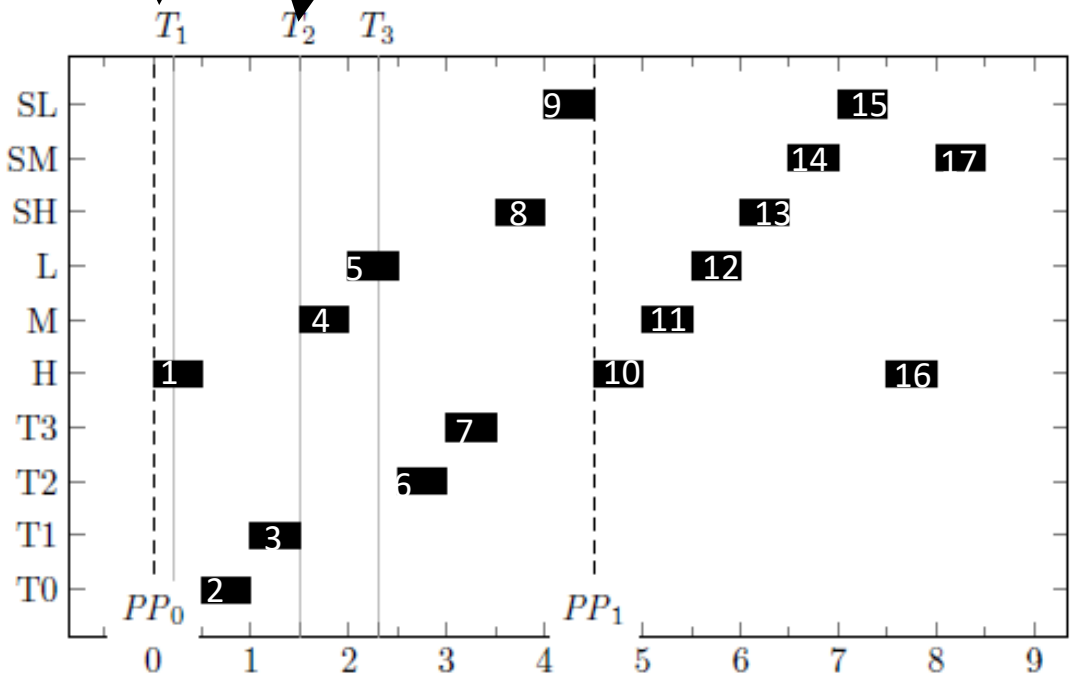
Effects of longer queues

L	M	H	SH	SL	L	M	H	SH	SL
1	2	3	4	5	6	7	8	9	10

5	4	1	8	9	12	11	10	13	15

SM	SM	H
11	12	13

14	17	16



See Casini, et al, ECRTS 2019, „Response-Time Analysis of ROS 2 Processing Chains“ for details. This applies to ROS 2 Crystal and Dashing.

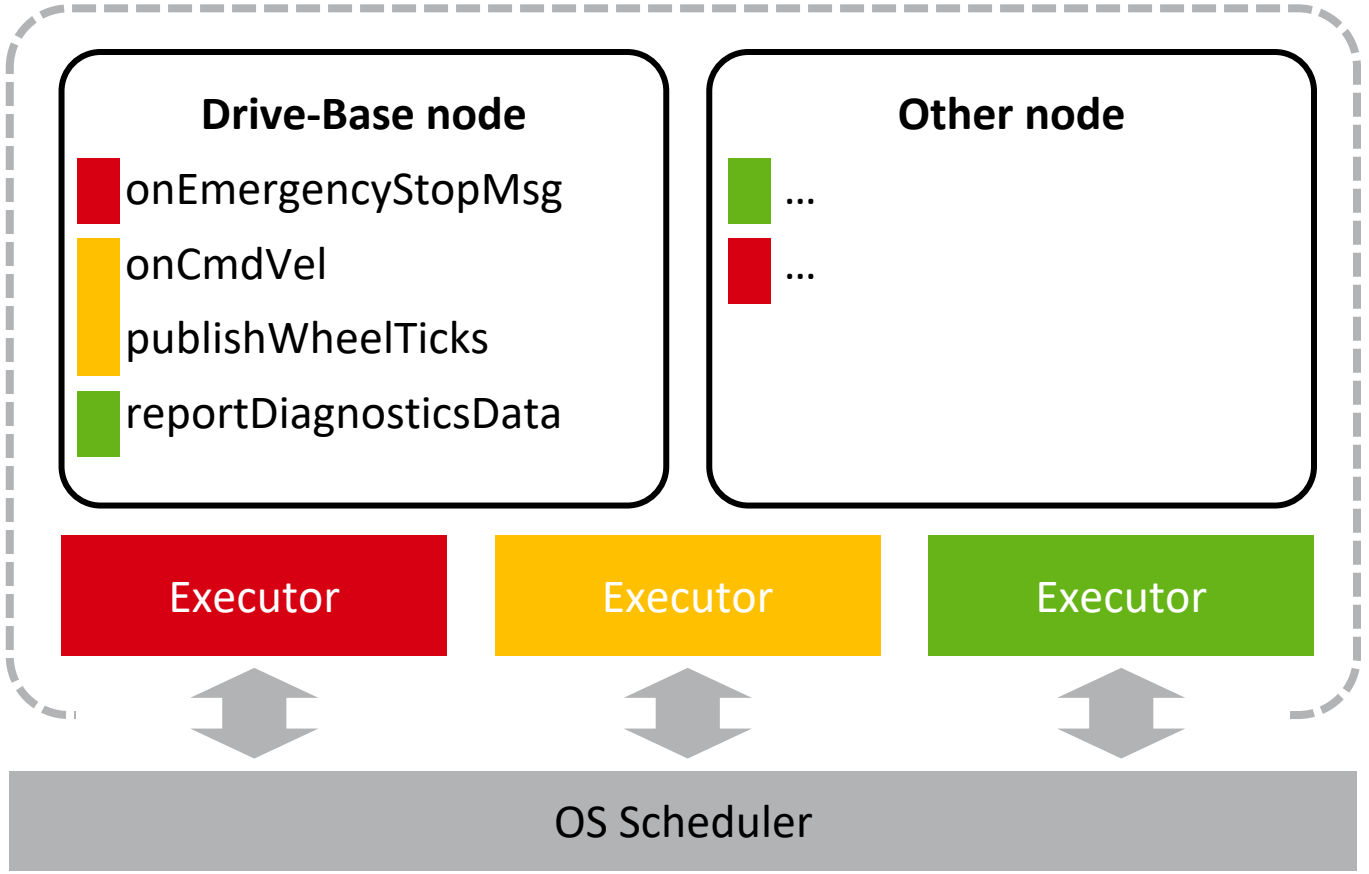
ROS 2 Execution: Determinism

What do we do?

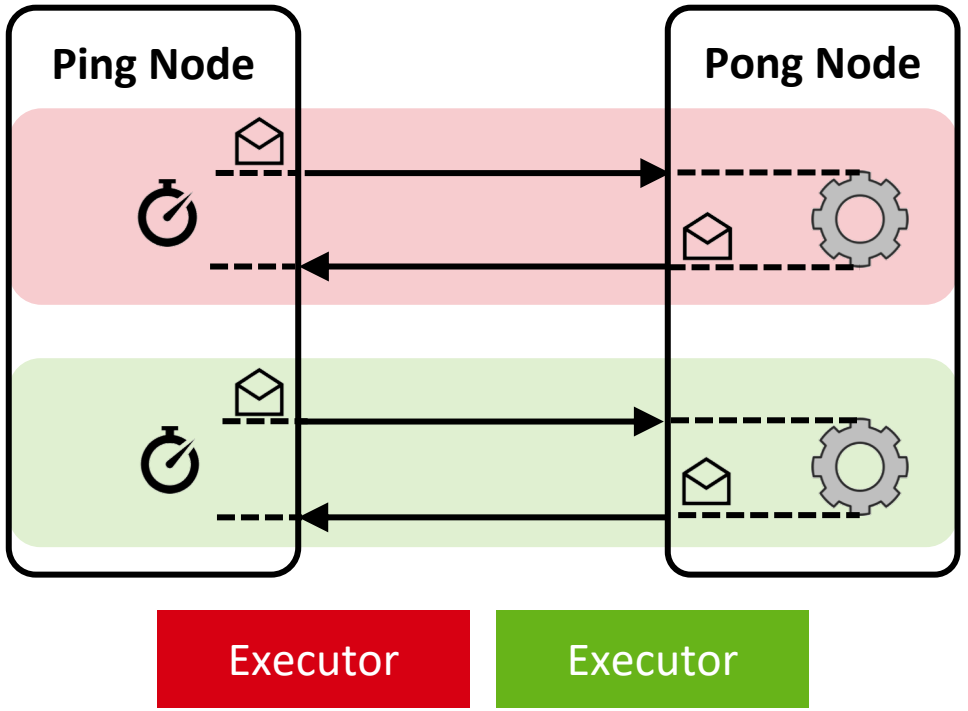
- Our current work
  - Improve the RMW API or the Executor/RMW work-split
- Coping mechanisms until then
  - Callback-group level executor
  - Architect your system around this

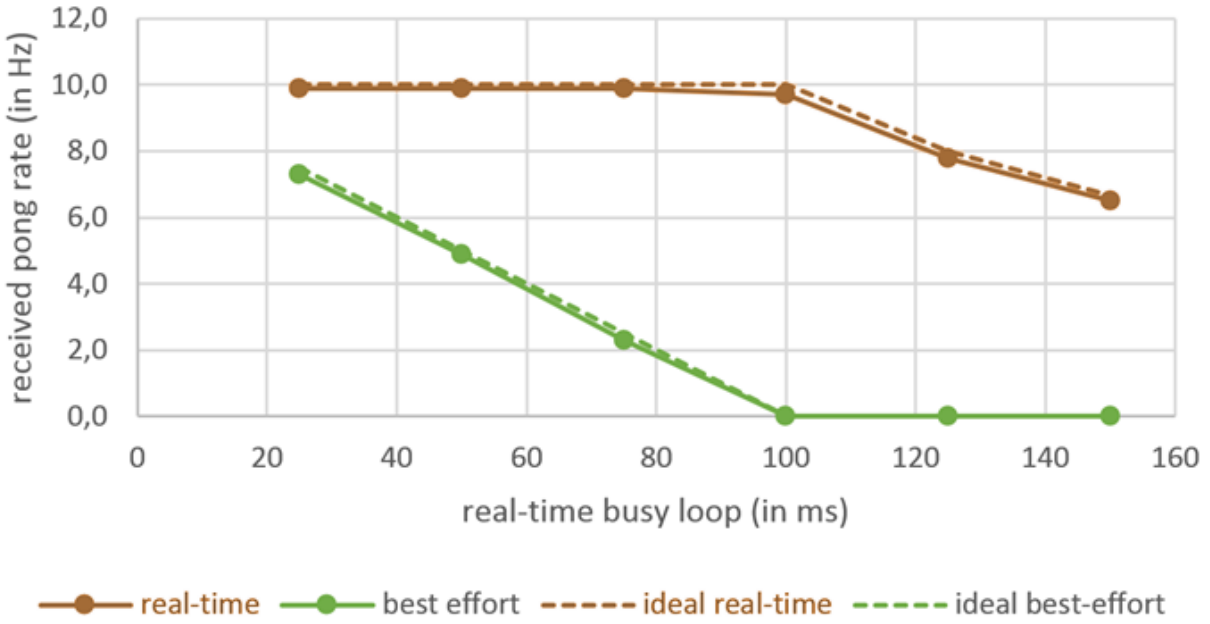
ROS 2 Execution

Callback-group level executor



ROS 2 Execution
Effect of priorities





github.com/boschresearch/ros2_examples → meta-executor

BOSCH

ROS 2 Execution: Determinism
Where do go from here?

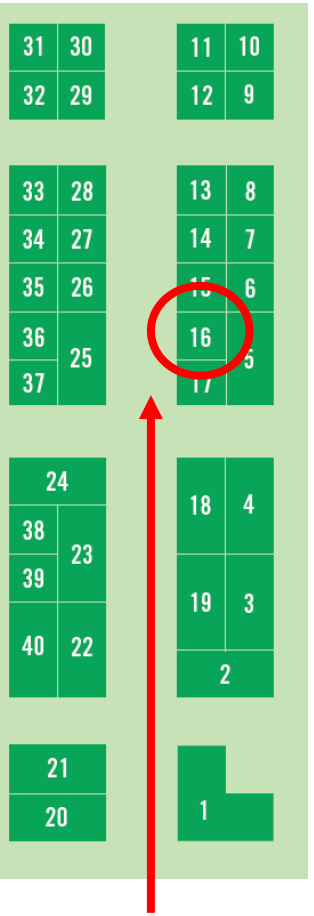
- We need a better RMW interface
  - See https://github.com/ros2/design/issues/259 for the current discussion
  - Essentially, we need more information than what's currently in the wait_set
- We need to update the communication status info *predictably*
  - For example, after every callback, or at regular intervals
  - It could always happen that something arrives which needs to be processed before the things we already know about
  - Right now, this is prohibitively expensive, however
- Maybe we also need different execution models to make things simple
  - For example, static order, or reactive execution (see Andrzej's talk), etc.

Please participate in the discussion!

ROS 2 Execution: Determinism

Contact

- E-Mail: ingo.luetkebohle@de.bosch.com
- Meet us at the Bosch booth
- Check out our related talks
  - Zero-Copy MW talk, Friday 14:00, Track 2
  - Micro-ROS talk on Friday, 14:40, Track 2
  - Many other relevant talks in track 2 as well



Bosch booth is #16.

We‘re there during breaks.

BOSCH

→ NOBLEO

BOSCH