Record and replay of ROS nodes with Mozilla rr
Deterministic debugging and reverse execution

Juraj Oršulić

University of Zagreb
Laboratory for Robotics and Intelligent Control Systems

juraj.orsulic@fer.hr

University of Zagreb

ROSCon 2018

FER
UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

LARICS

LABORATORY FOR ROBOTICS AND INTELLIGENT CONTROL SYSTEMS
Department of Control and Computer Engineering

Mozilla rr - record and replay

moz://a

- process execution recording ● REC
  - **lightweight, low performance overhead**
    - only **non-deterministic** inputs are recorded
  - can record entire process trees (e.g. multiple ROS nodes)
- process execution replay ⏯ ⏮ ⏭
  - execution flow is reconstructed
    - using the same binary executable and the recorded non-deterministic inputs
    - **instruction-level** replay accuracy
  - a gdbserver interface is used to debug the replayed process

rr design

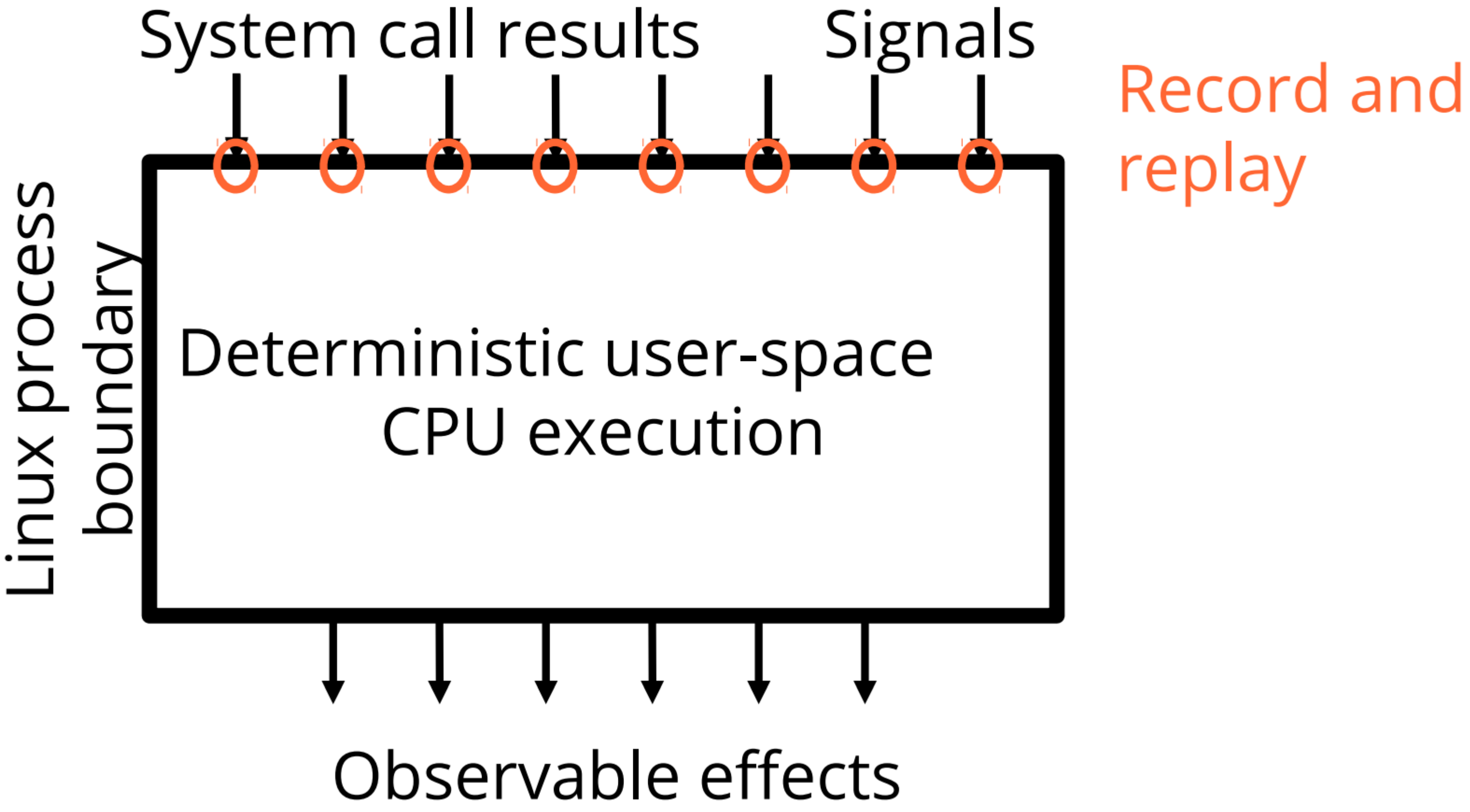
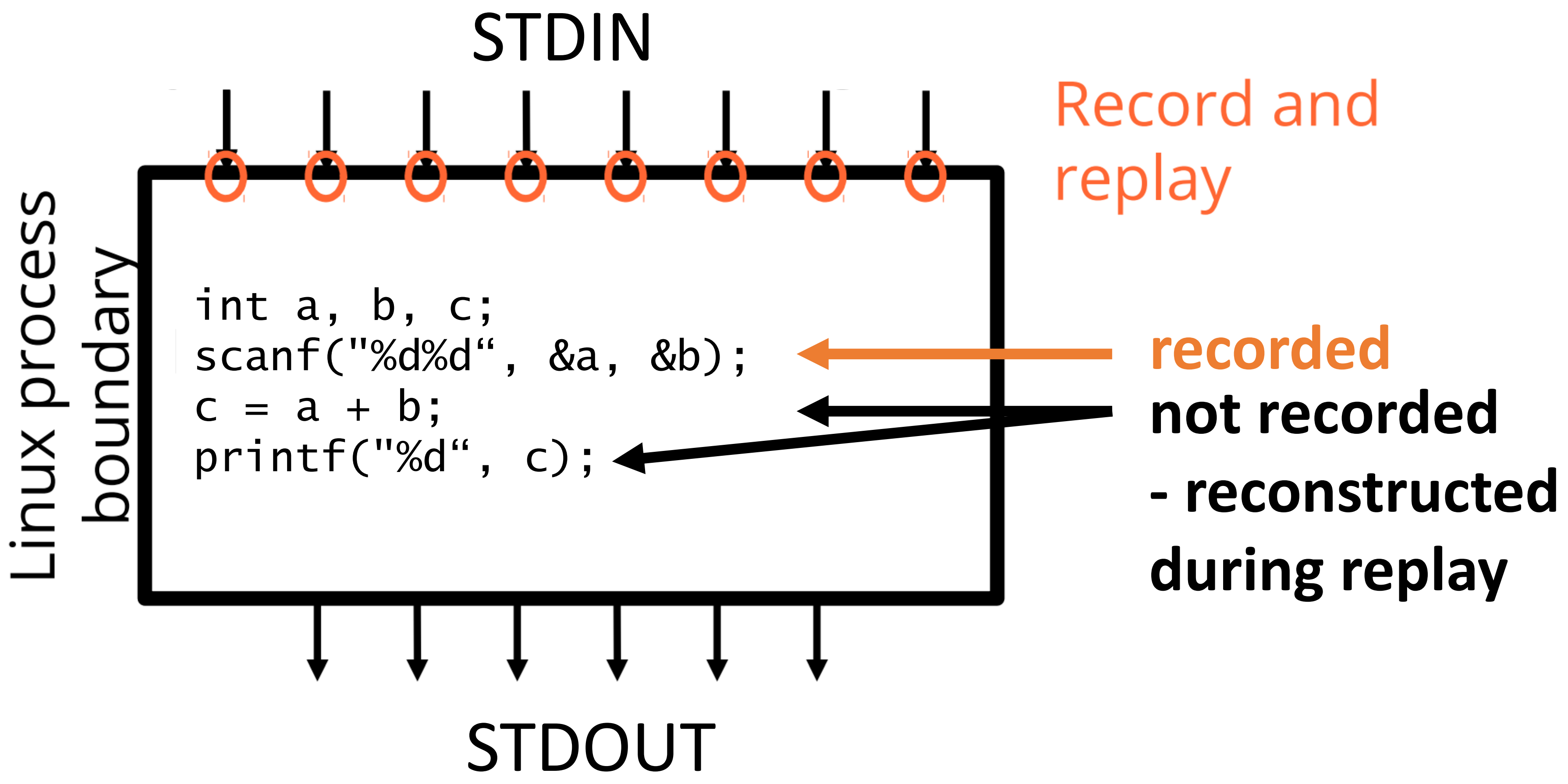


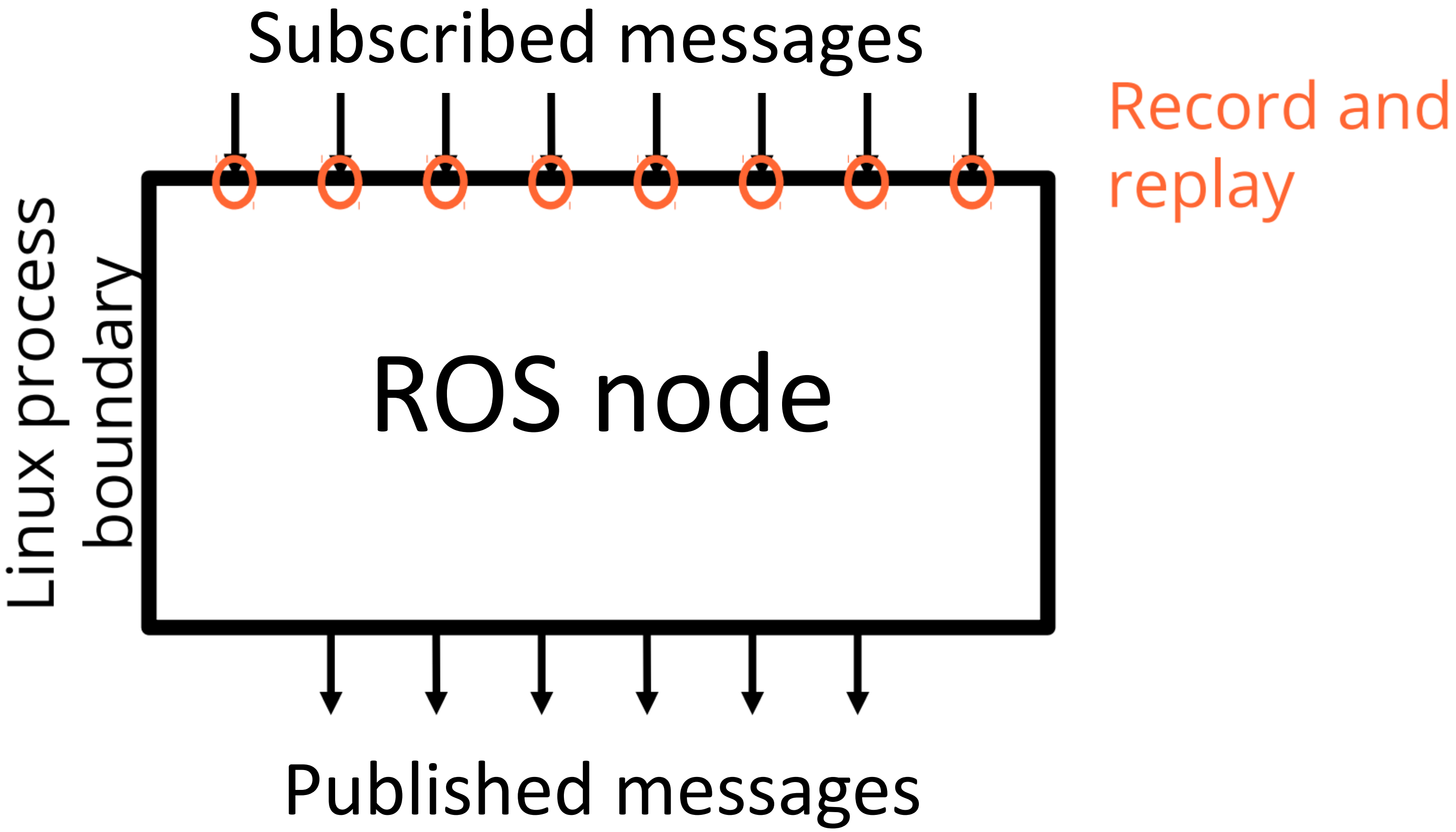
Diagram from Robert O'Callahan, “Practical Record And Replay Debugging With rr”

rr design

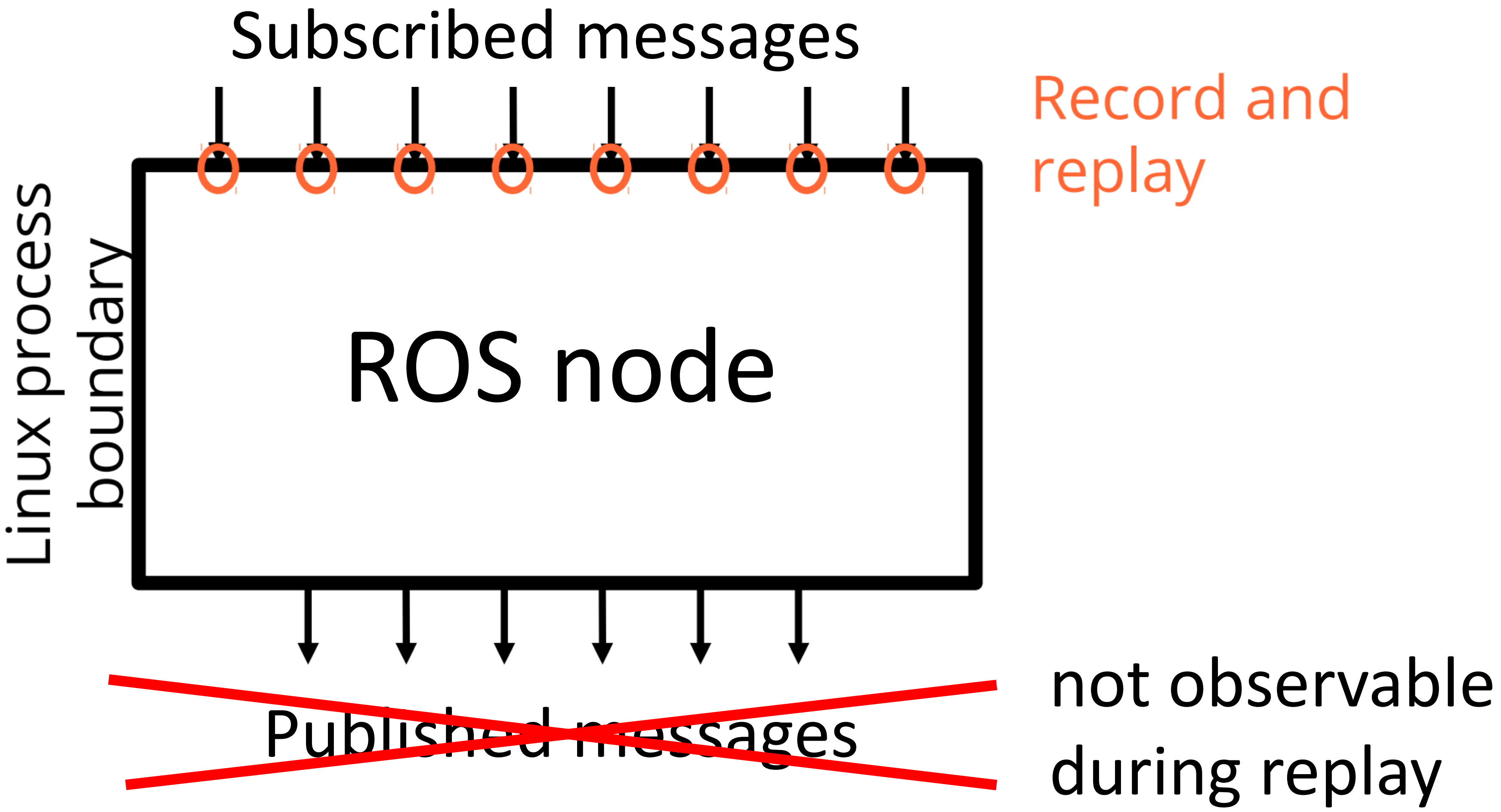


```
int a, b, c;
scanf("%d%d“, &a, &b);
c = a + b;
printf("%d“, c);
```

rr design


Subscribed messages
Record and replay
Linux process boundary
ROS node
Published messages

rr design



Subscribed messages

Record and replay

Linux process boundary

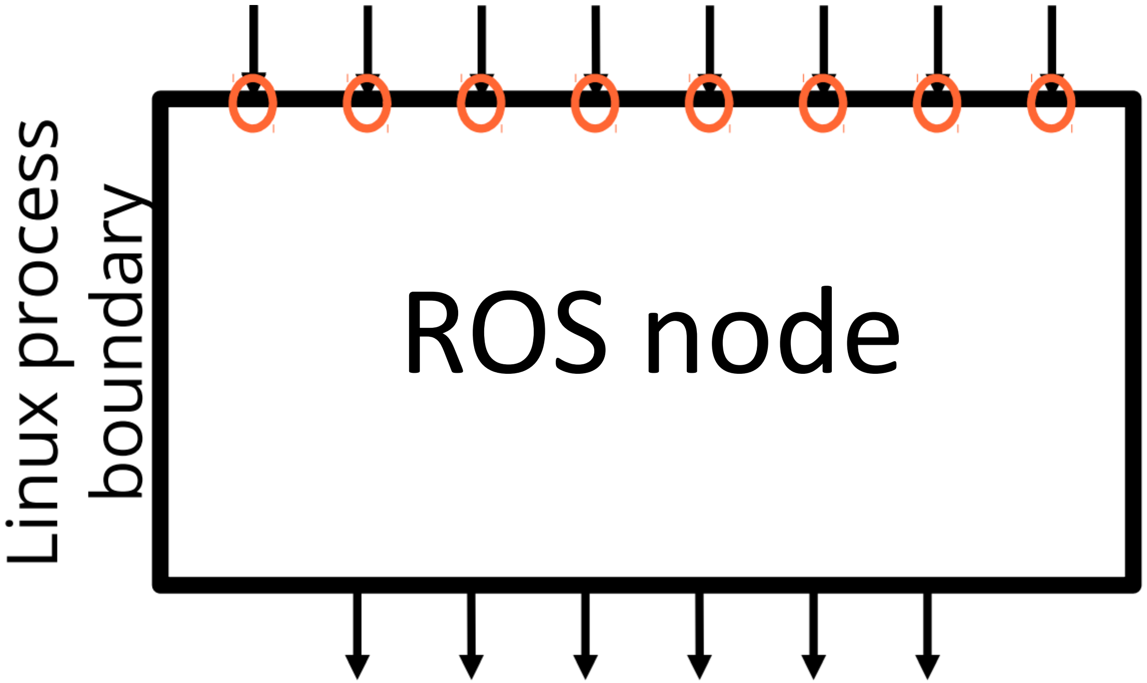
ROS node

~~Published messages~~

not observable during replay

rr design

Subscribed messages



Linux process boundary

ROS node

Record and replay

Published messages

republish with rospy + gdb Python hooks

Caveats

- only for Linux on x86(_64)
  - Intel (>= ~Sandy Bridge)
  - recently merged experimental AMD support

- multithreaded processes are executed sequentially (on a single core) -> slower
  - rr supports a chaotic thread scheduling mode,
    useful for provoking race condition bugs

Ideas for the future (call for action?)

- republishing ROS messages
  - although documented, currently a proof-of-concept hacky script
  - engage with upstream rr for proper integration (a side effect reconstruction plugin?)

- (long-term) recording on robots?

Thank you for your attention!

- `juraj.orsulic@fer.hr`
- `apt install rr` on Ubuntu (git master is better)
- https://rr-project.org/

- https://github.com/larics/gdb_ros_publisher
- http://larics.fer.hr/laricswiki/doku.php?id=software:debugging

- demo video:
  https://youtu.be/tC6ggFehems