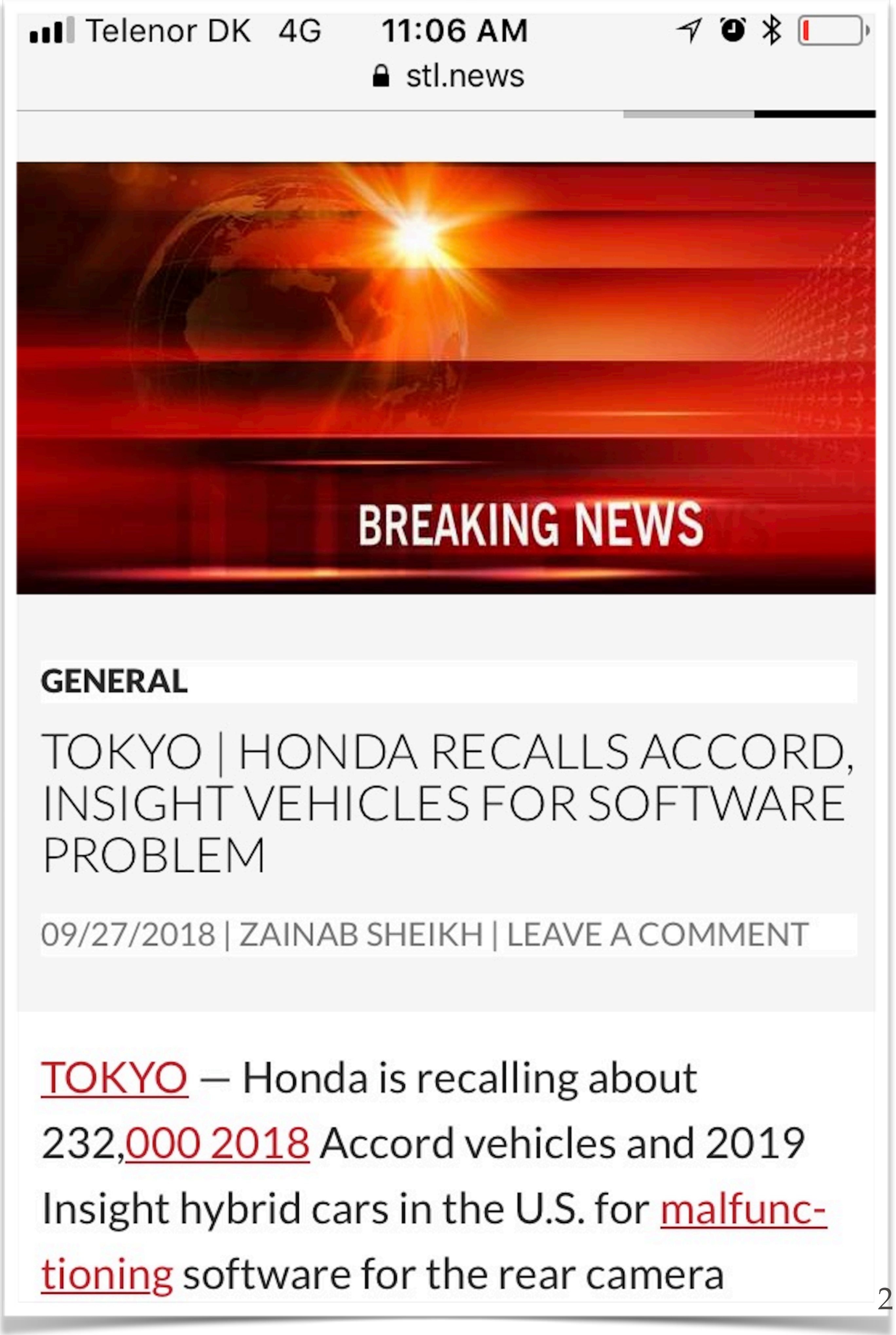
rosin-project.eu

Quality Assurance Initiatives for ROS Tool Development (Part II)

Zhoulai Fu
Sep 29, 2018
IT University of Copenhagen



GENERAL

TOKYO | HONDA RECALLS ACCORD, INSIGHT VEHICLES FOR SOFTWARE PROBLEM

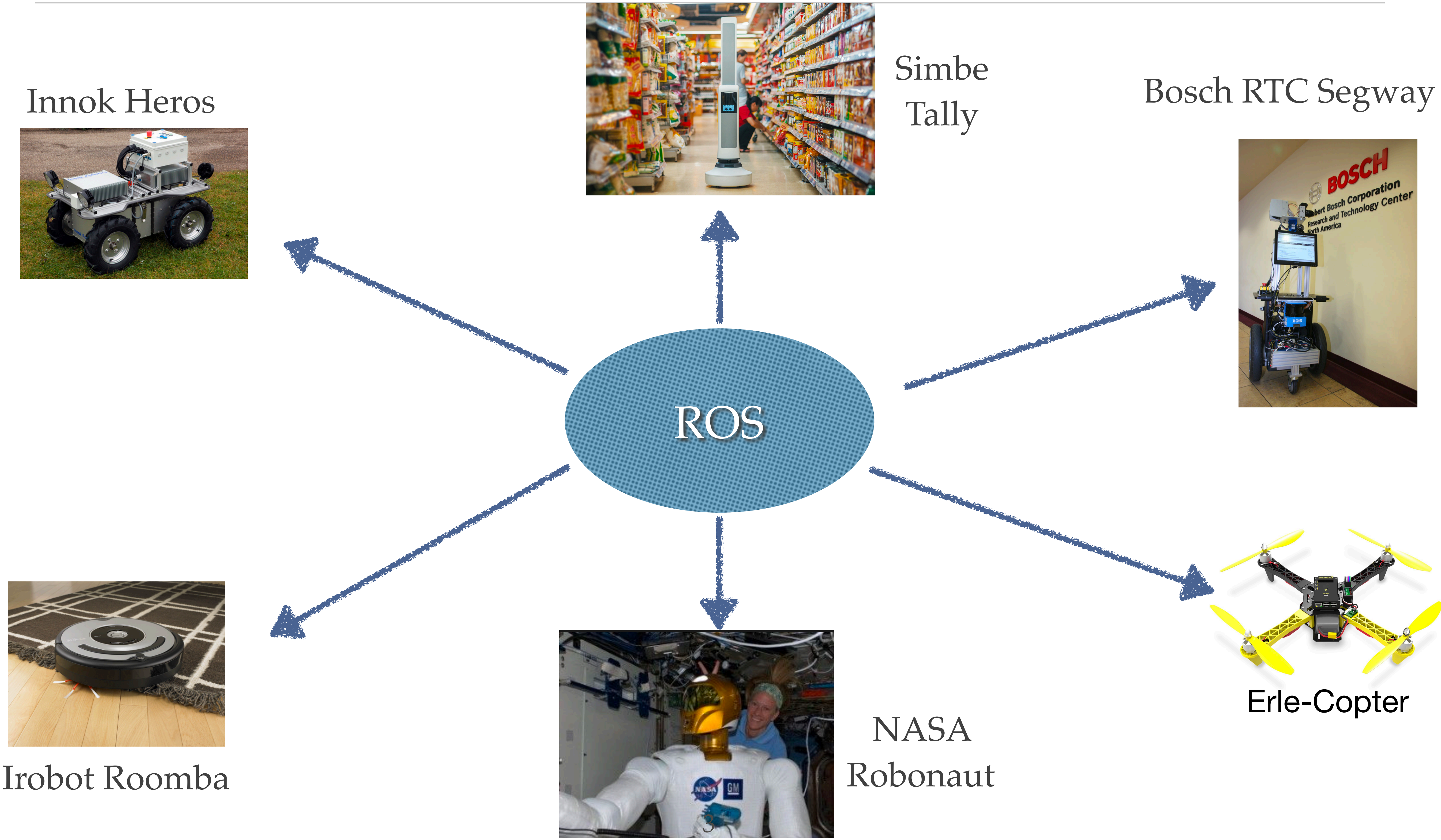
09/27/2018 | ZAINAB SHEIKH | LEAVE A COMMENT

TOKYO — Honda is recalling about 232,000 2018 Accord vehicles and 2019 Insight hybrid cars in the U.S. for malfunctioning software for the rear camera

Software

Reliability

An Important Goal: Making ROS (more) Reliable



ROS

Simbe Tally

Innok Heros

Bosch RTC Segway

Irobot Roomba

NASA Robonaut

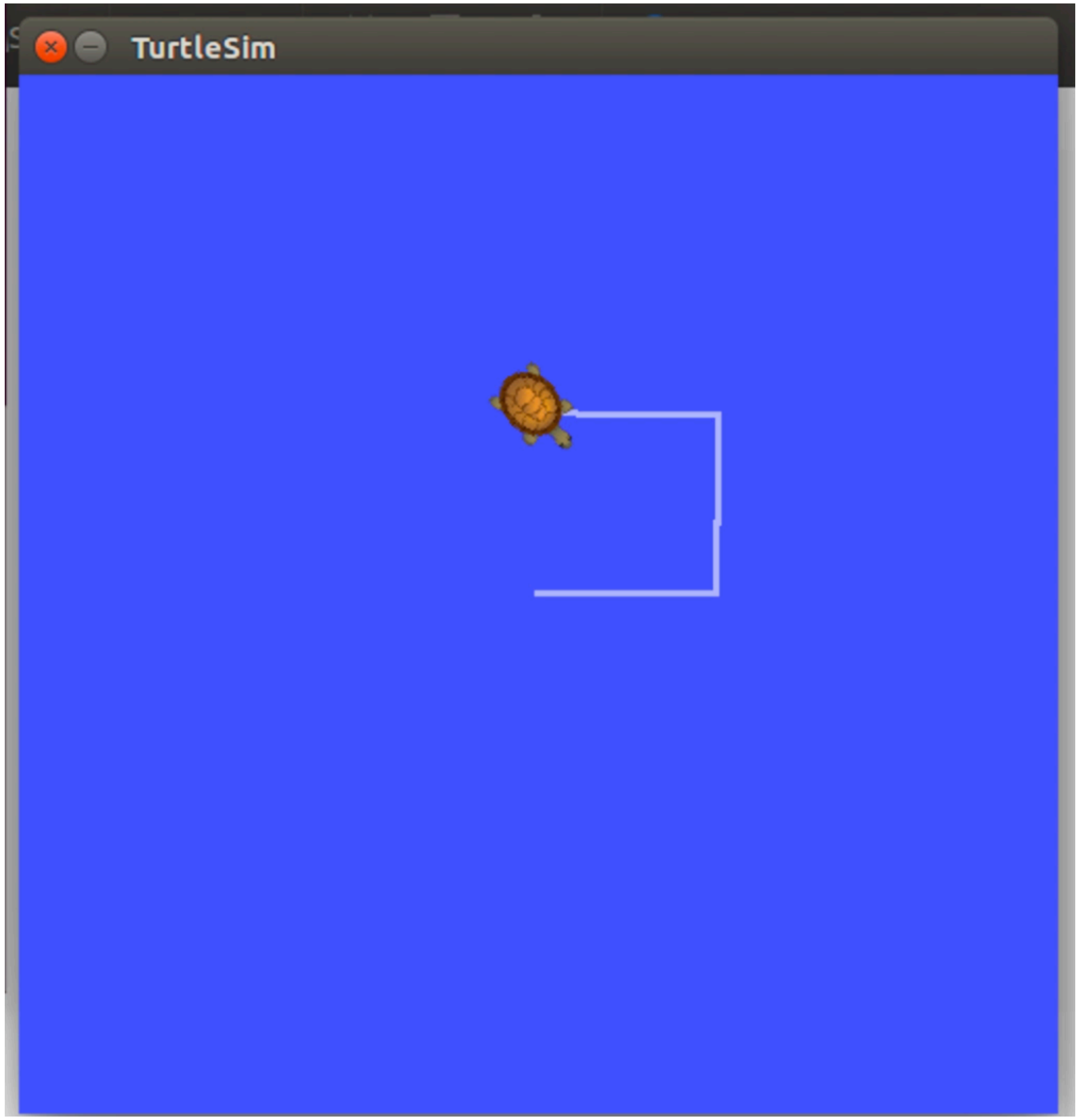
Erle-Copter

Since 2017, we have been working on how to test ROS automatically

This talk:

- Dealing with ROS-specific challenges
- Reusing existing tools
- Demo from our on-going work

Challenge 1 in testing ROS: Lack of specification



- Turtlesim package

- Expected: draw square

- Bug: Turtle spins after 3/4

ROS bugs often occur "**silently**"

Challenge 2 in testing ROS: Lack of “good” test drivers

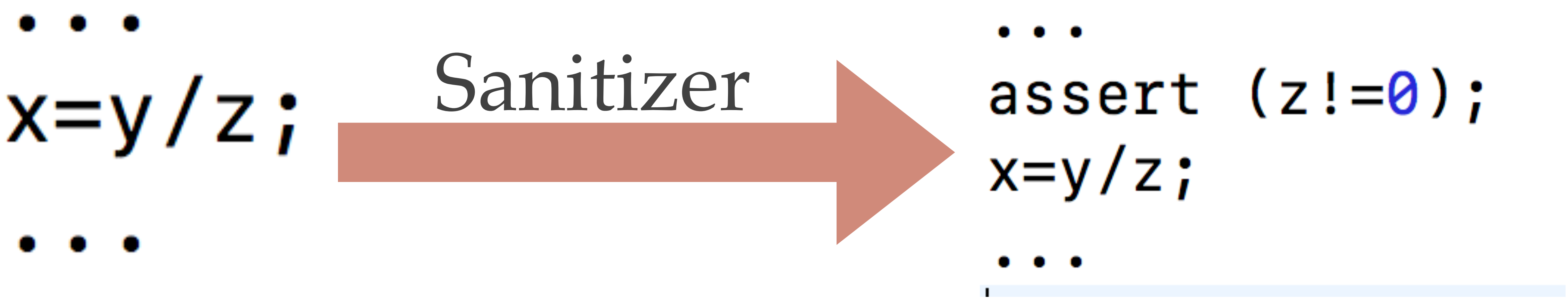
- Test drivers launch ROS components
- *Good* test drivers fail ROS components

Our solution: If you don’t have it, try to get it

- Challenge 1: Lack of specification ==> Use *sanitizer*
- Challenge 2: Lack of good test drivers ==> Use *fuzzing*

Existing, mature stuff

What is sanitizer?



```
...
x=y/z;
...
```

Sanitizer

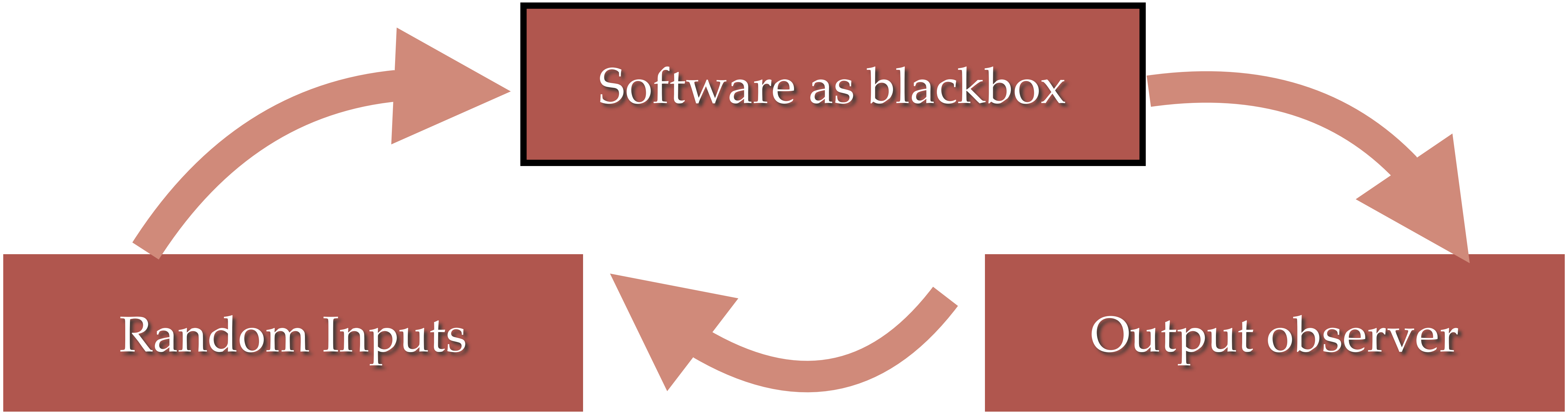
```
...
assert (z!=0);
x=y/z;
...
```

- A **build-in** compiler option in GCC and Clang
- **Automatically** inject assertions: division-by-zero, array index out of bound

Demo available offline

What is fuzzing?

- 1989 experiment from Univ. Wisconsin
    - Pure random testing crashed **1/4** of the tested Unix utilities
- Fuzzing = **Smart** random testing

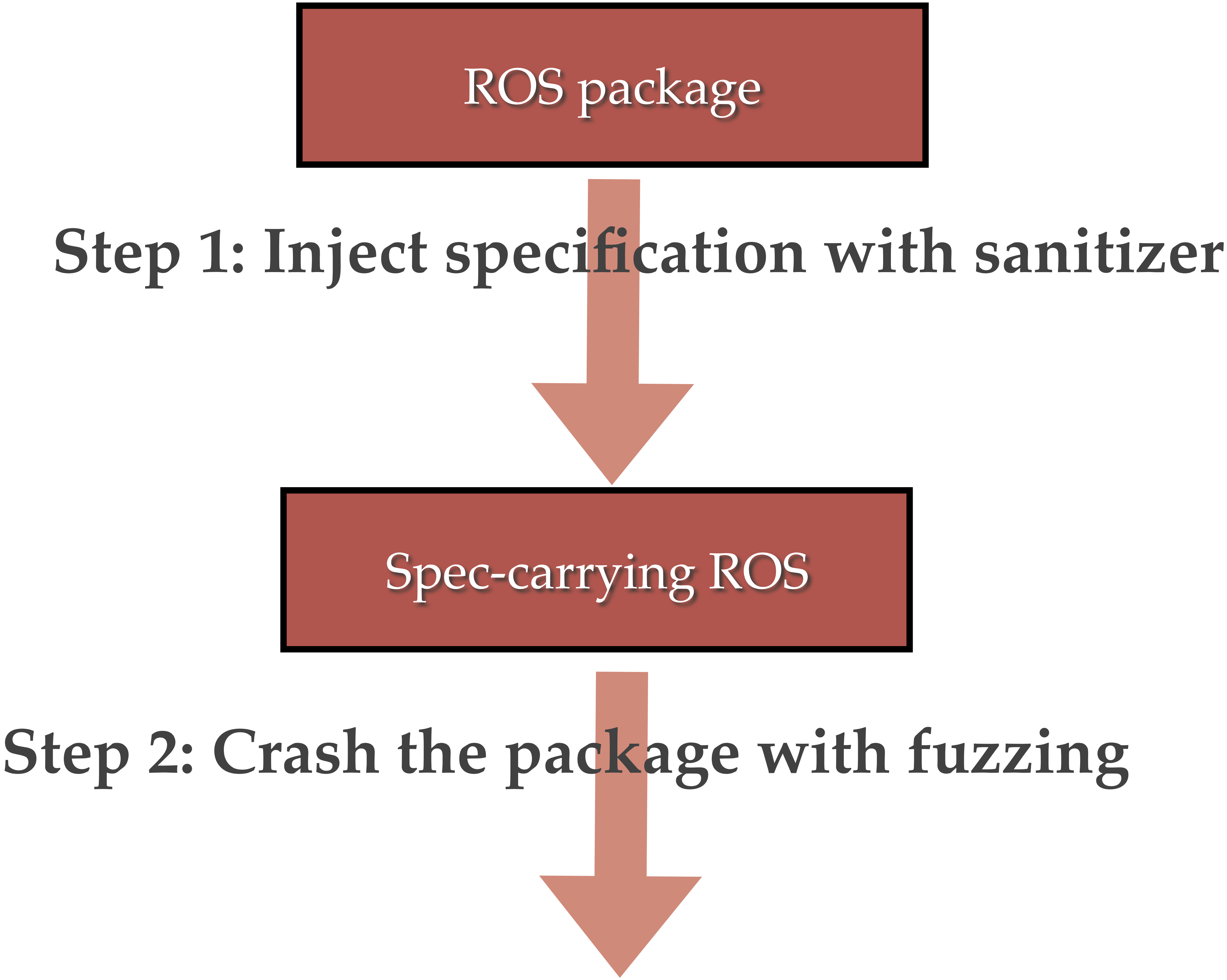


Software as blackbox

Random Inputs

Output observer

- Numerous bugs detected by fuzzing
- Microsoft, Google and many companies use fuzzing **daily**



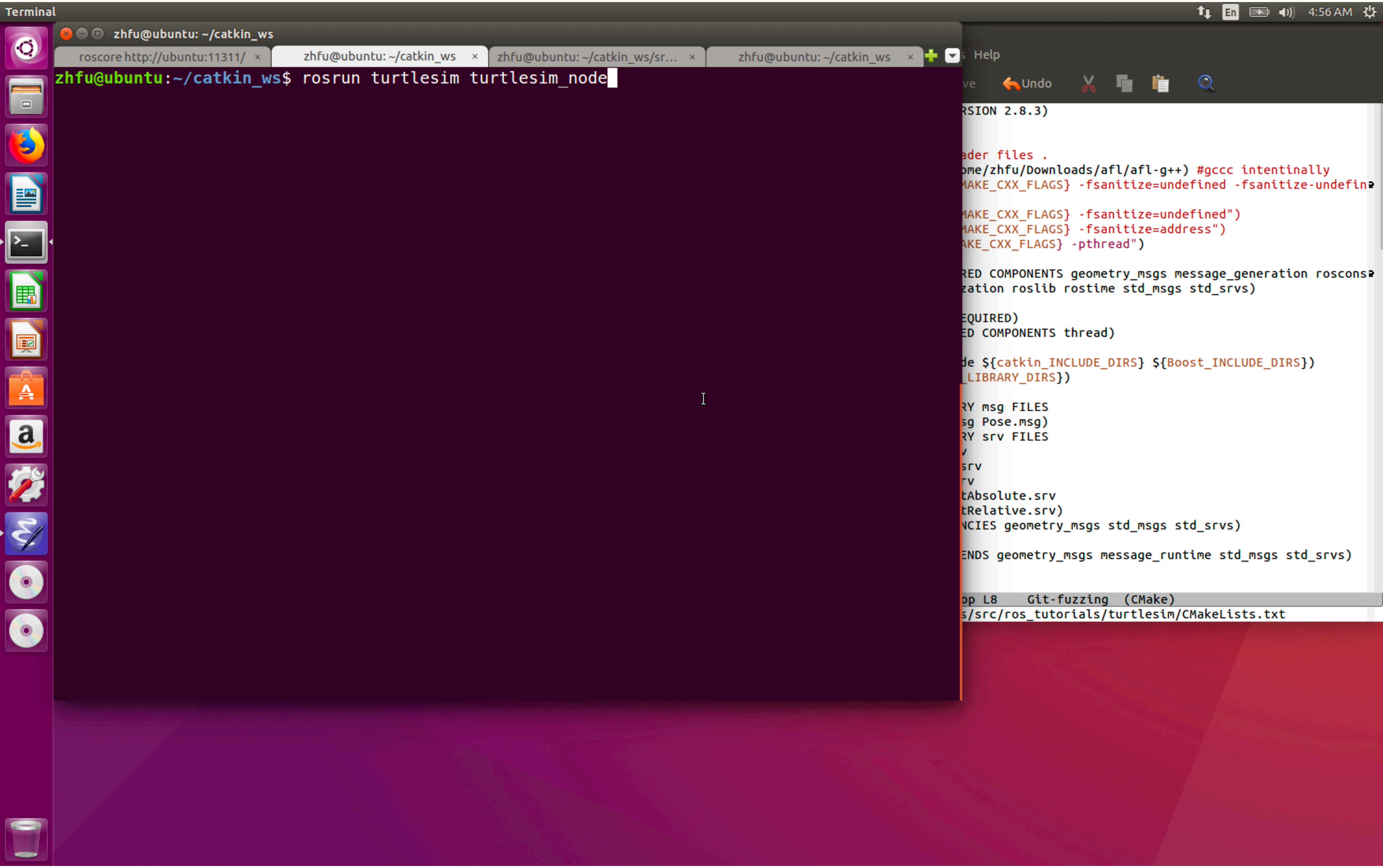
ROS package

Step 1: Inject specification with sanitizer

Spec-carrying ROS

Step 2: Crash the package with fuzzing

Demo from our on-going work



```
zhfu@ubuntu:~/catkin_ws$ rosrun turtlesim turtlesim_node
```

```
RSION 2.8.3)


ader files .
ome/zhfu/Downloads/afl/afl-g++) #gccc intentinally
MAKE_CXX_FLAGS} -fsanitize=undefined -fsanitize-undefin

MAKE_CXX_FLAGS} -fsanitize=undefined")
MAKE_CXX_FLAGS} -fsanitize=address")
AKE_CXX_FLAGS} -pthread")

RED COMPONENTS geometry_msgs message_generation roscons
zation roslib rostime std_msgs std_srvs)

EQUIRED)
ED COMPONENTS thread)

de ${catkin_INCLUDE_DIRS} ${Boost_INCLUDE_DIRS})
_LIBRARY_DIRS})

RY msg FILES
sg Pose.msg)
RY srv FILES
v
srv
rv
tAbsolute.srv
tRelative.srv)
NCIES geometry_msgs std_msgs std_srvs)

ENDS geometry_msgs message_runtime std_msgs std_srvs)

op L8    Git-fuzzing  (CMake)
s/src/ros_tutorials/turtlesim/CMakeLists.txt
```

Conclusion on Tool Development part

- Build **reliable** ROS components
- We **reuse** existing solutions: sanitizer + fuzzing

Thank you! Questions?

Meet us at ROS-Industrial booth #10

ROSin

rosin-project.eu