Understanding Misconfigurations in ROS: An Empirical Study and Current Approaches

Paulo Canelas

with Bradley Schmerl, Alcides Fonseca, and Christopher S. Timperley

Carnegie Mellon University

University of Lisbon

International Symposium on Software Testing and Analysis (ISSTA). 2024.

Understanding Misconfigurations in ROS: An Empirical Study and Current Approaches

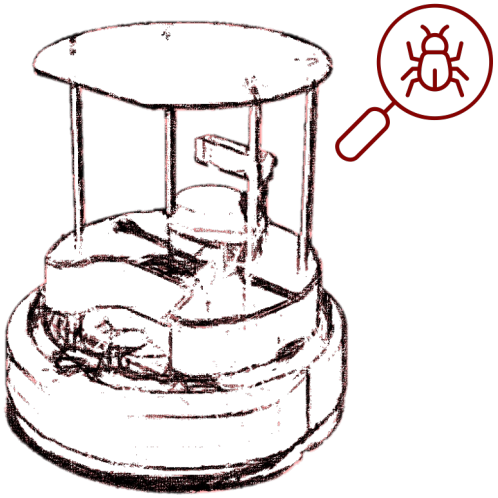
Paulo Canelas

with Bradley Schmerl, Alcides Fonseca, and Christopher S. Timperley

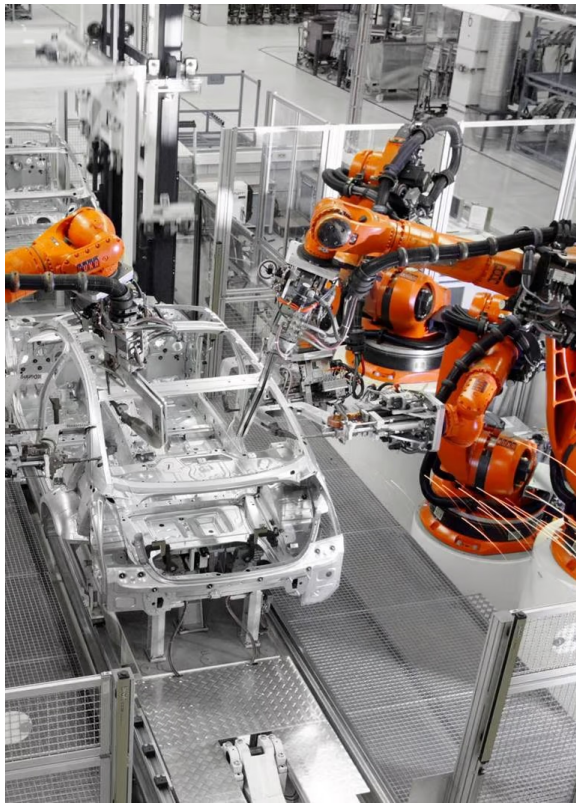
Carnegie Mellon University

University of Lisbon

International Symposium on Software Testing and Analysis (ISSTA). 2024.



Robotic systems are playing a critical role in today's society by performing a wide range of tasks



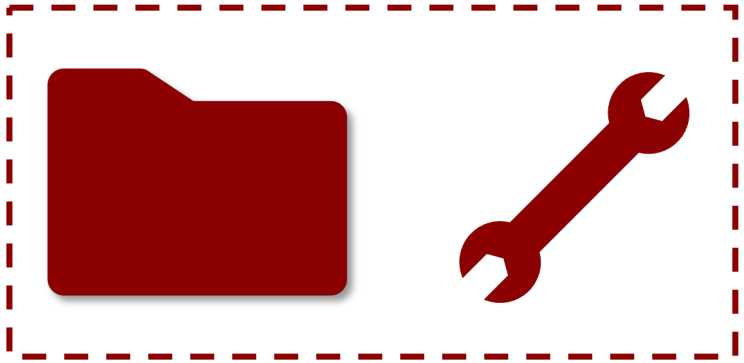
Automotive Industry

Autonomous Vehicles

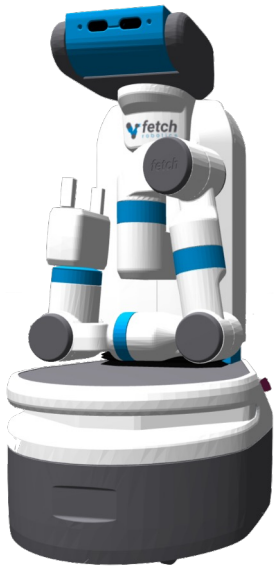
Drone Delivery

The Robot Operating System (ROS) improves robotics development by providing reusable components

"We have designed ROS to support our **philosophy of modular**, tools-based software development"
[Quigley et al, 2009]



Libraries and tools available for composing



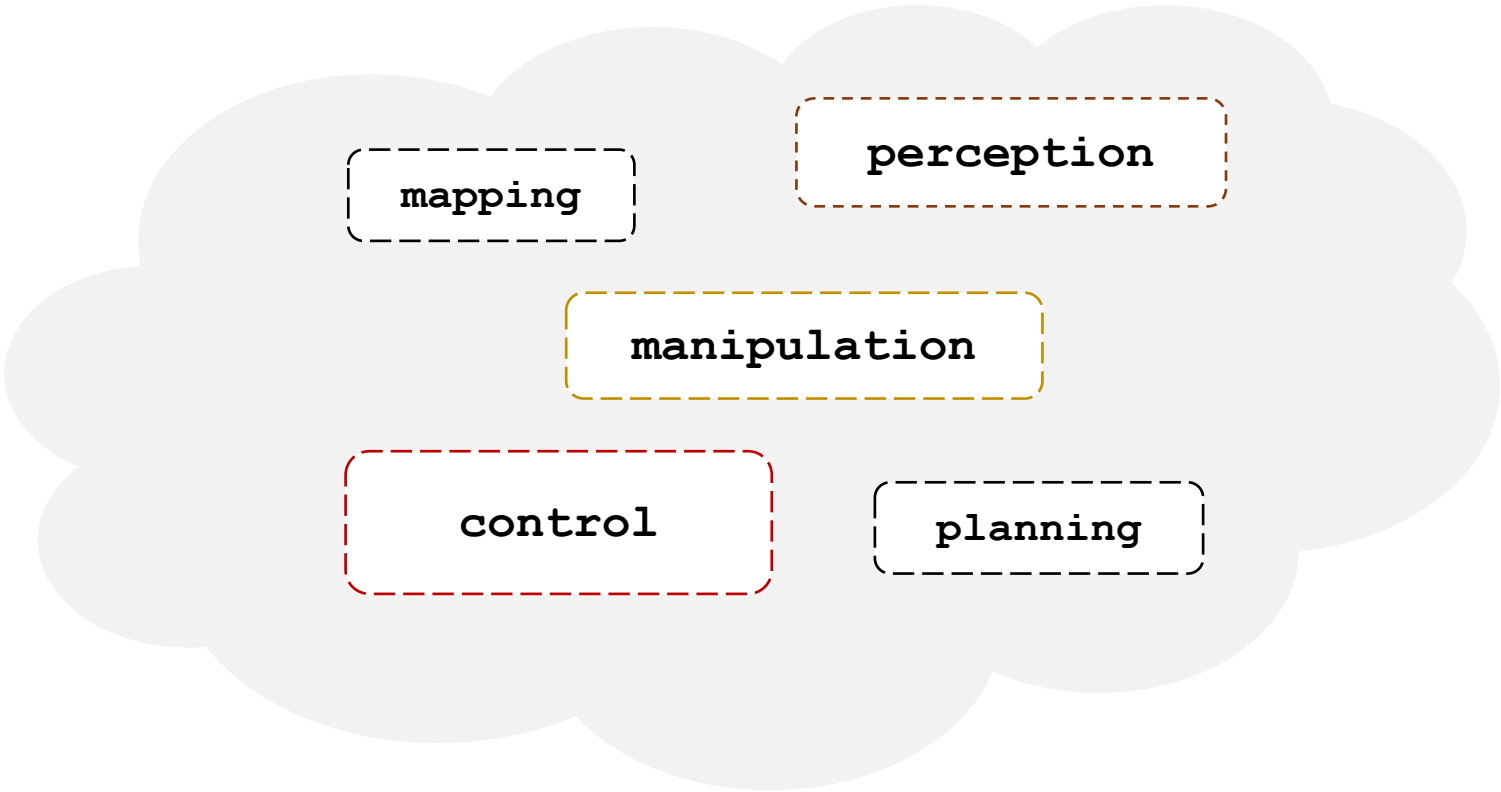
Quickly prototype the robotic system

BOSCH

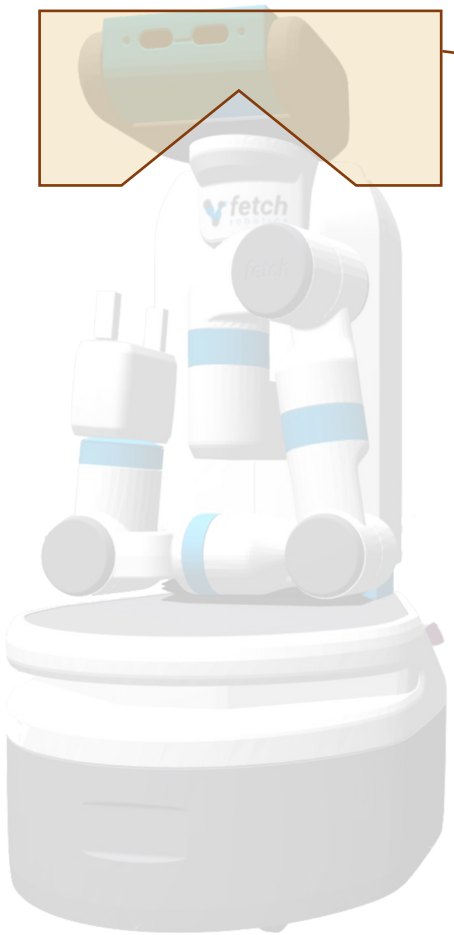
BOEING

Popular adoption in the industry

ROS allows developers to configure and integrate reusable, off-the-shelf components



ROS allows developers to configure and integrate reusable, off-the-shelf components



perception

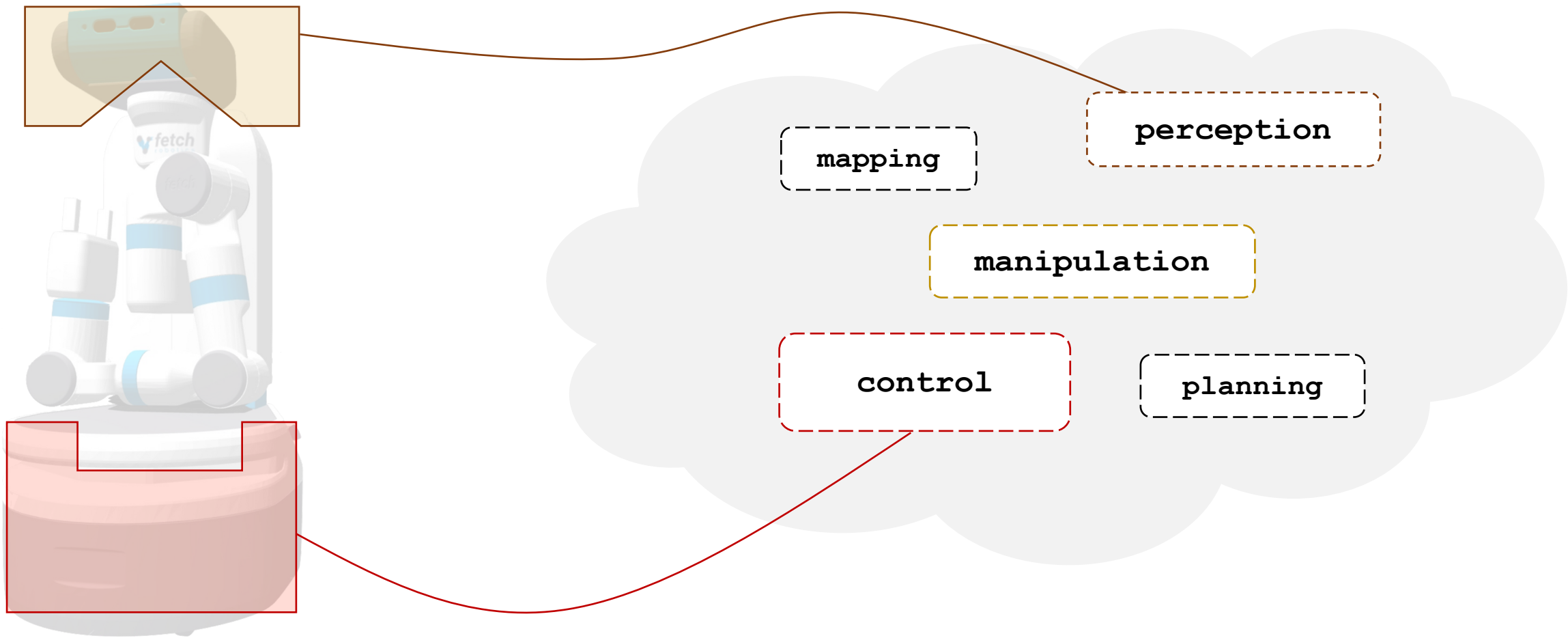
mapping

manipulation

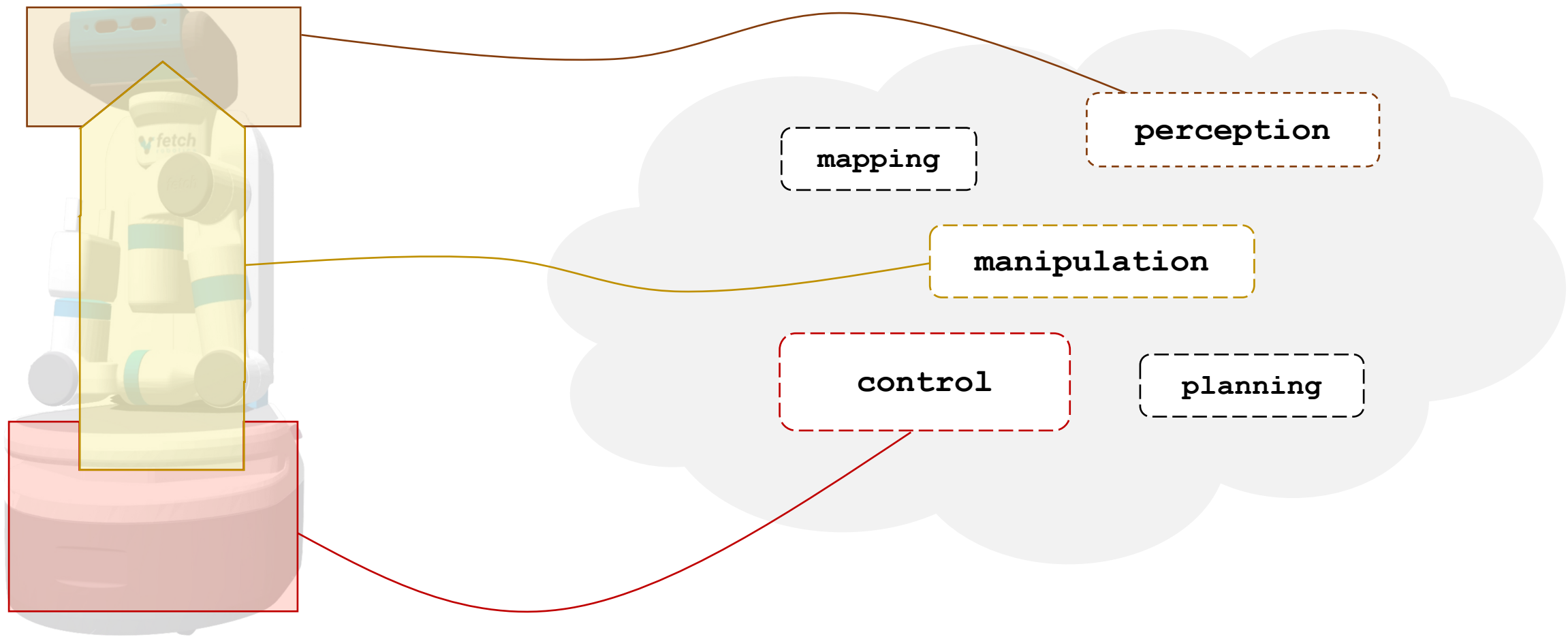
control

planning

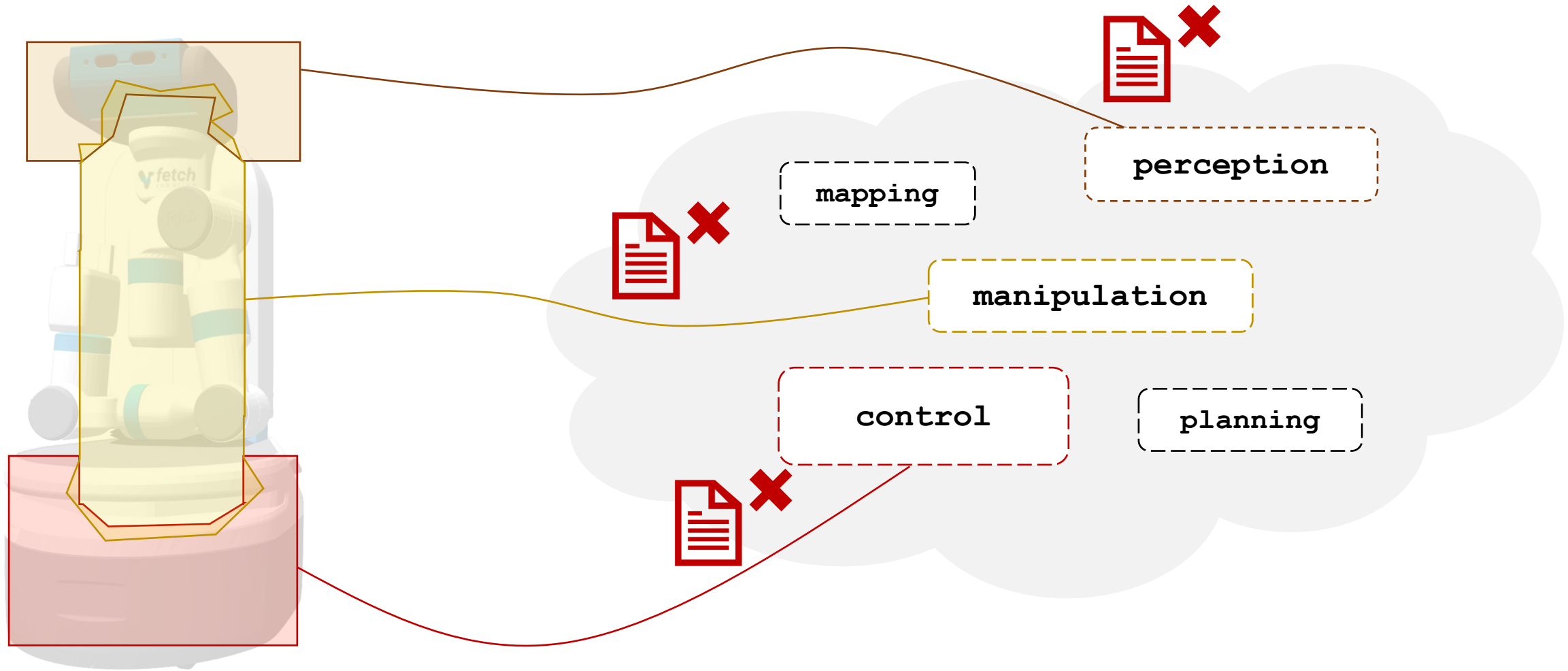
ROS allows developers to configure and integrate reusable, off-the-shelf components



ROS allows developers to configure and integrate reusable, off-the-shelf components



However, configuration is not trivial and the lack of documentation in components leads to errors

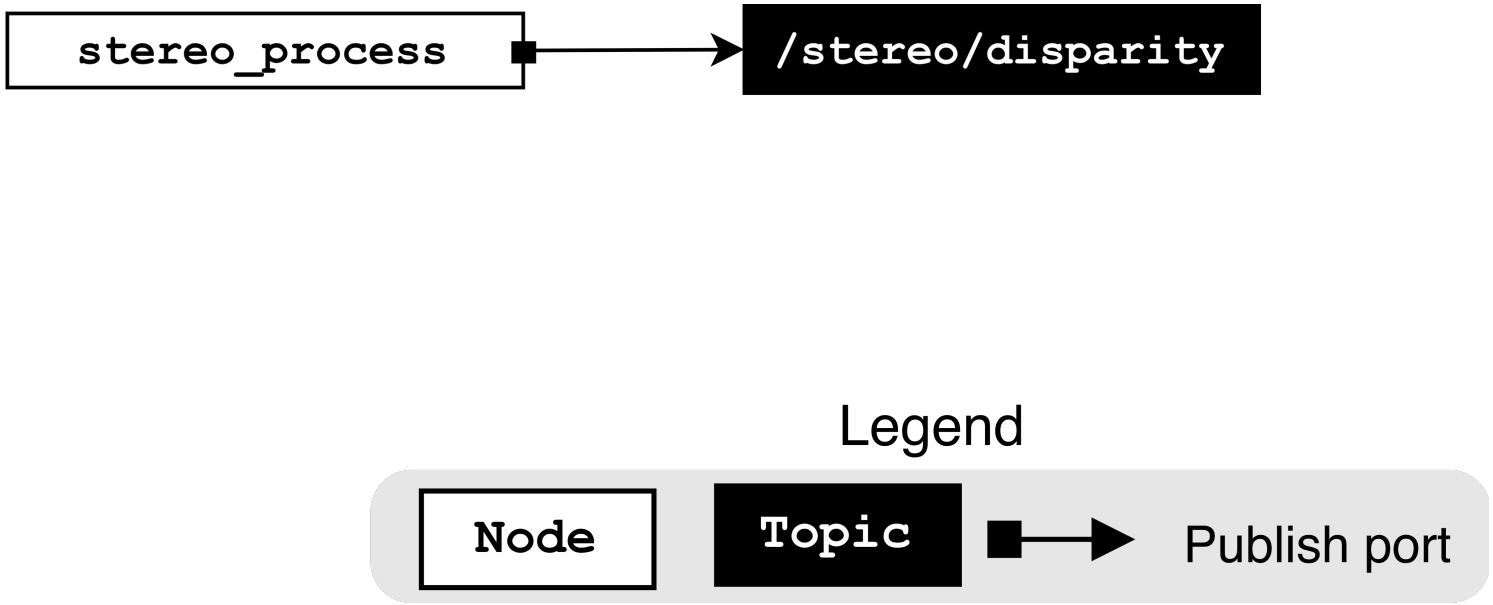


In this work, we understand the broader set of configuration errors in ROS and what techniques address them

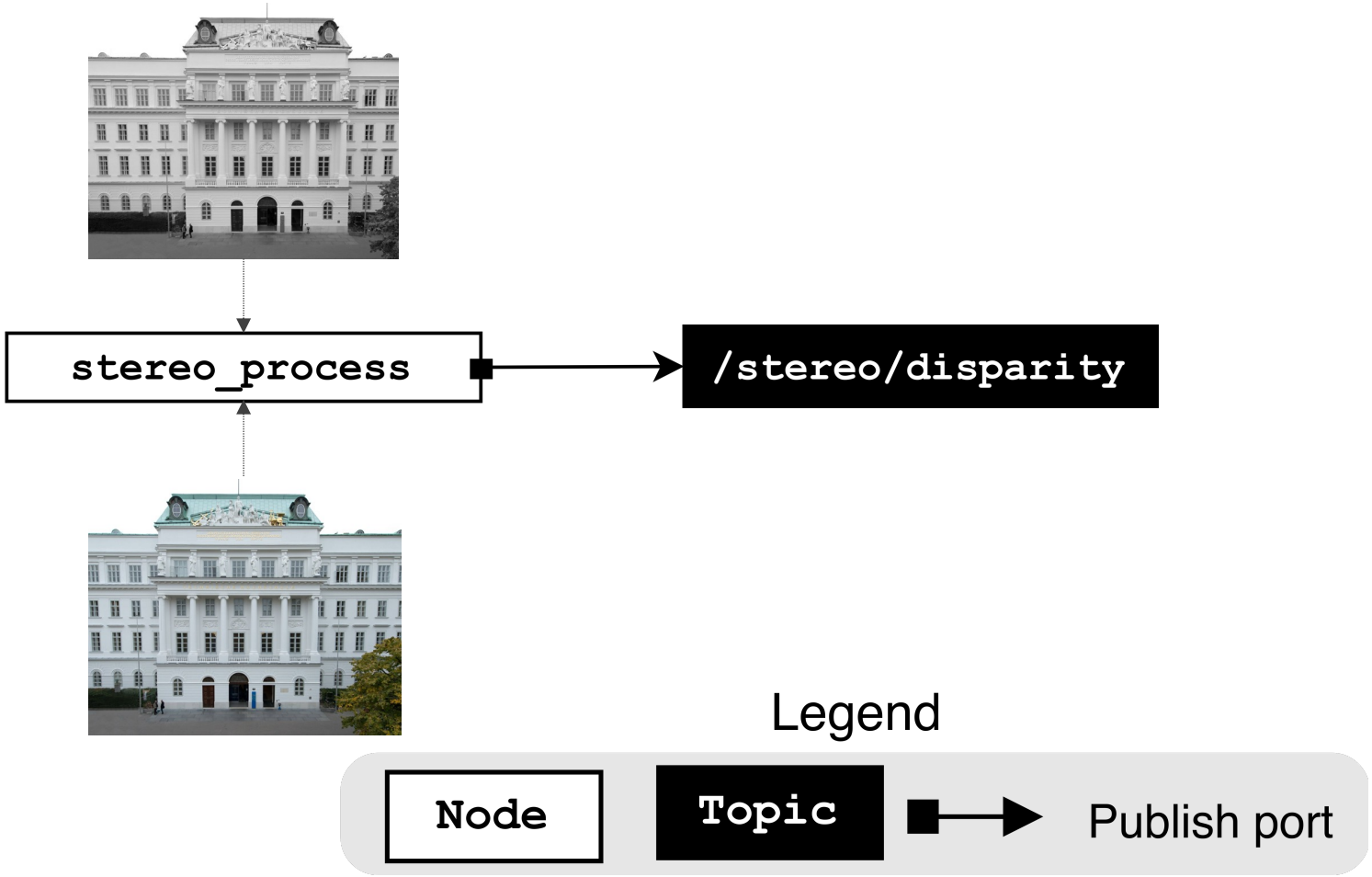
In this work, we understand the broader set of configuration errors in ROS and what techniques address them

Spoiler Alert: *we found 50 different types of configuration errors, many of which no analysis technique can detect!*

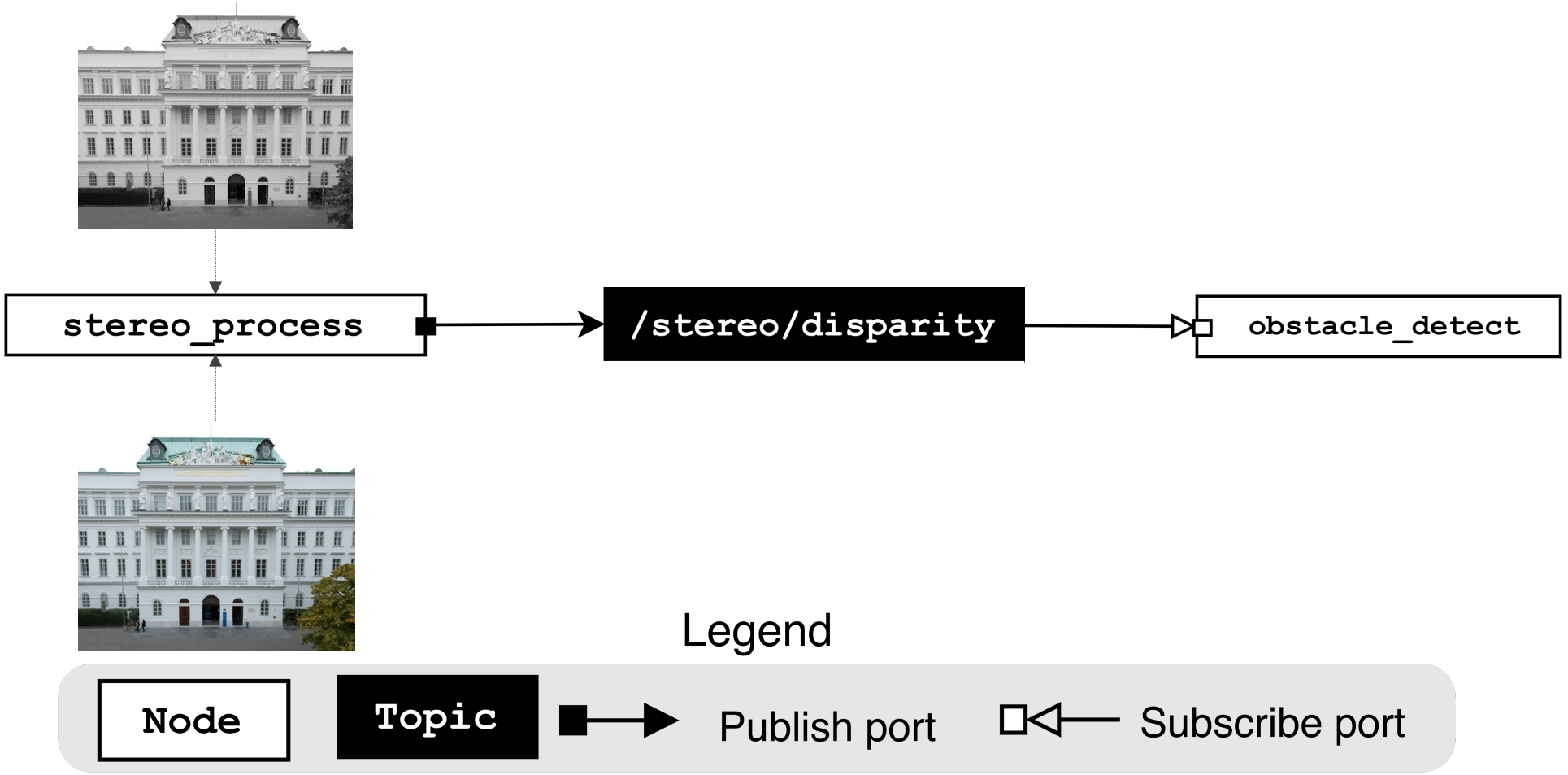
In ROS, components process inputs and may produce an output (e.g., nodes in publisher-subscriber)



In ROS, components process inputs and may produce an output (e.g., nodes in publisher-subscriber)

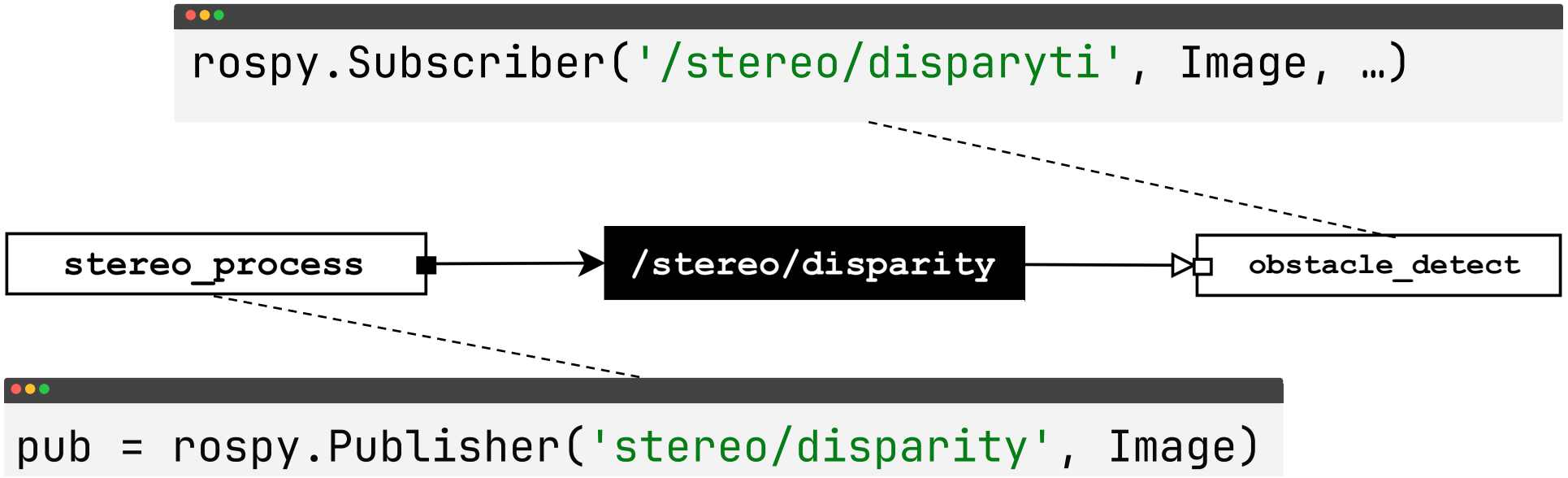


In ROS, components process inputs and may produce an output (e.g., nodes in publisher-subscriber)

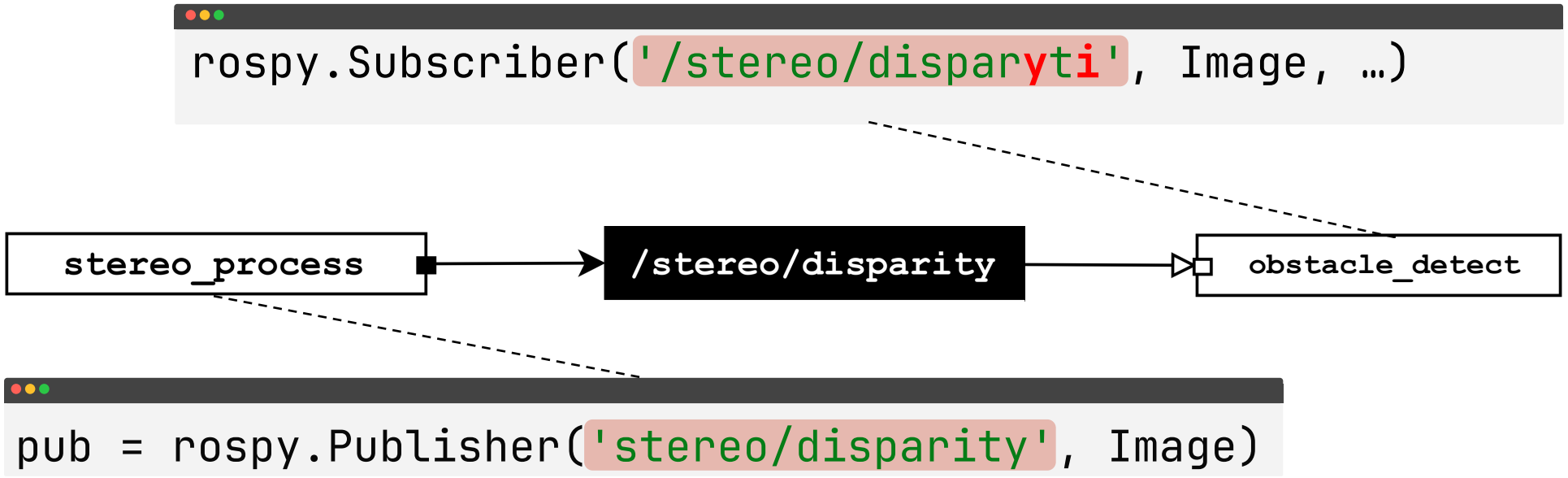


It is up to developers to ensure that components assumptions match and the system is well configured

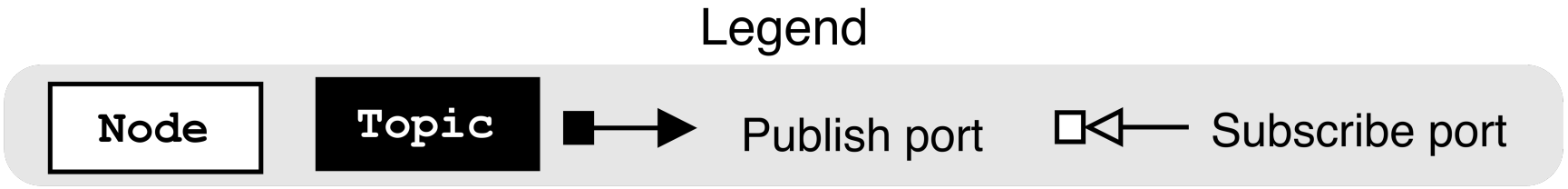
It is up to developers to ensure that components assumptions match and the system is well configured



It is up to developers to ensure that components assumptions match and the system is well configured



Misconfigurations arise when the architecture does not match developers expectations

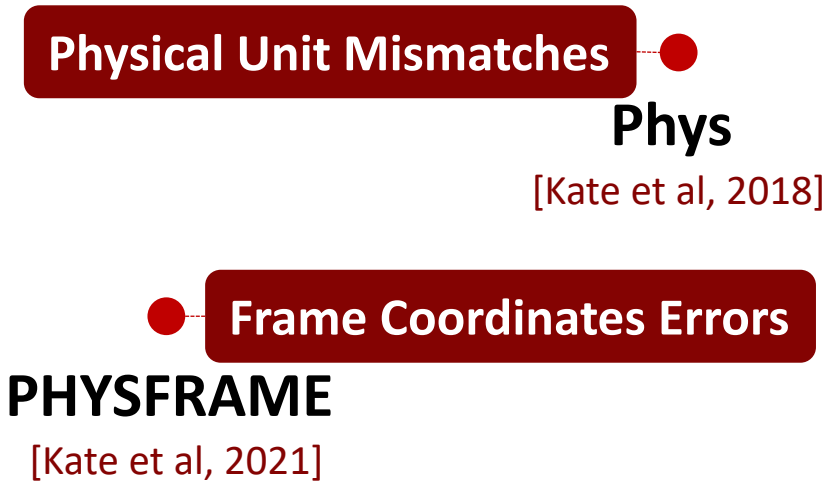


Prior work addressed specific well-known categories of misconfigurations in ROS systems

Prior work addressed specific well-known categories of misconfigurations in ROS systems

[Kate et al, 2018]

Prior work addressed specific well-known categories of misconfigurations in ROS systems



Prior work addressed specific well-known categories of misconfigurations in ROS systems

Physical Unit Mismatches

Phys

[Kate et al, 2018]

Frame Coordinates Errors

PHYSFRAME

[Kate et al, 2021]

Structural Misconfigurations

HAROS [Santos et al, 2021]

ROSDiscover [Timperley et al, 2022]

Prior work addressed specific well-known categories of misconfigurations in ROS systems

Behavioral Misconfigurations

ROSInfer
[Dürschmid et al, 2024]

Physical Unit Mismatches

Phys
[Kate et al, 2018]

Frame Coordinates Errors

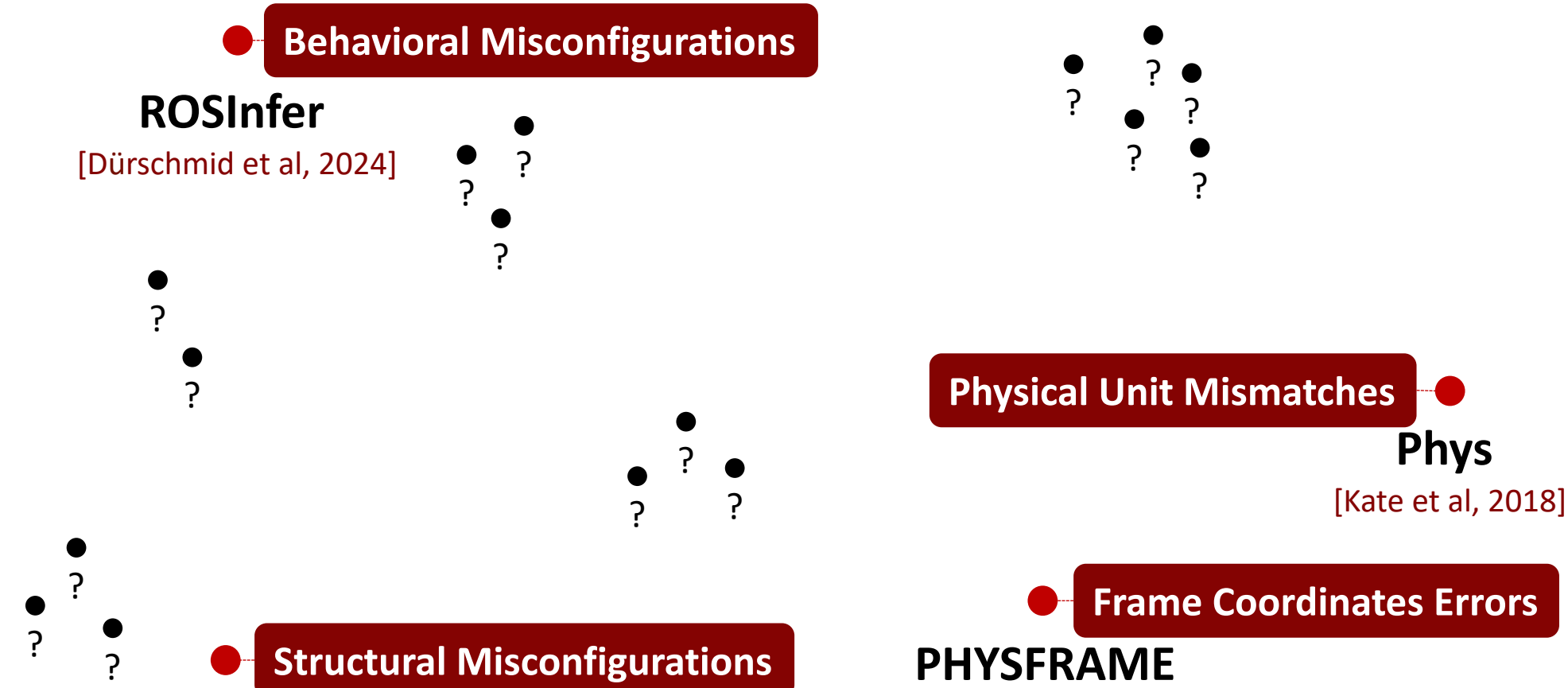
PHYSFRAME
[Kate et al, 2021]

Structural Misconfigurations

HAROS [Santos et al, 2021]

ROSDiscover [Timperley et al, 2022]

However, to effectively detect these errors we must understand the broader set of misconfigurations



We study the broader set of misconfigurations to identify the gap in analysis tools in detecting them

We study the broader set of misconfigurations to identify the gap in analysis tools in detecting them

RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

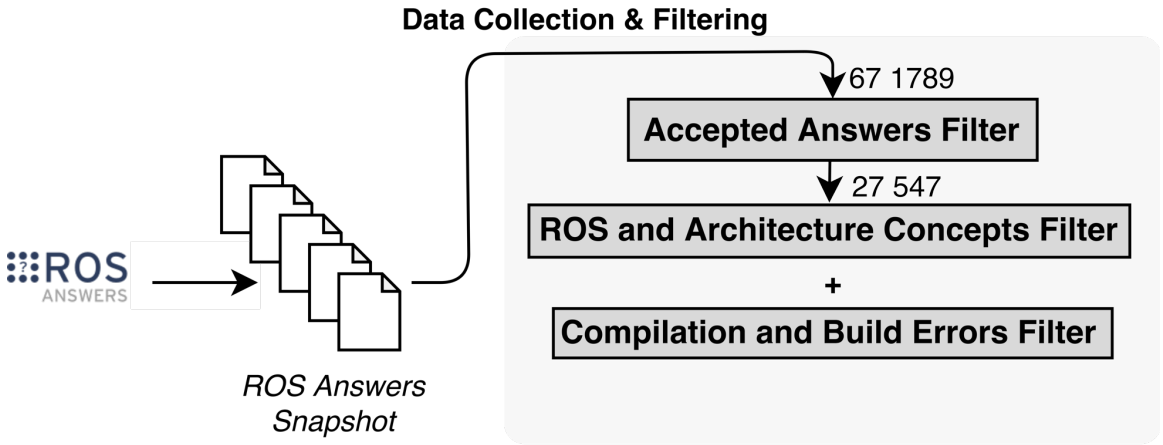
RQ2. To what extent do current techniques address these categories of misconfiguration?

We study the broader set of misconfigurations to identify the gap in analysis tools in detecting them

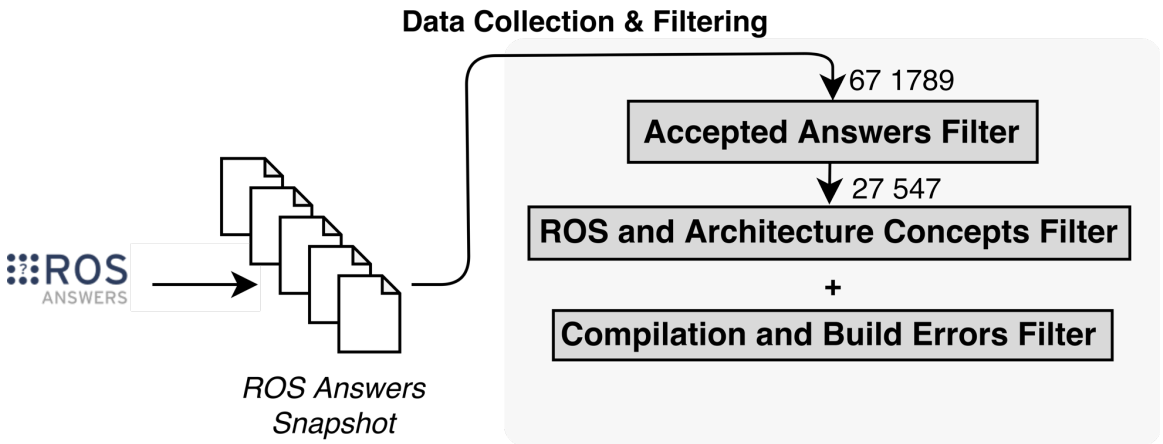
RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

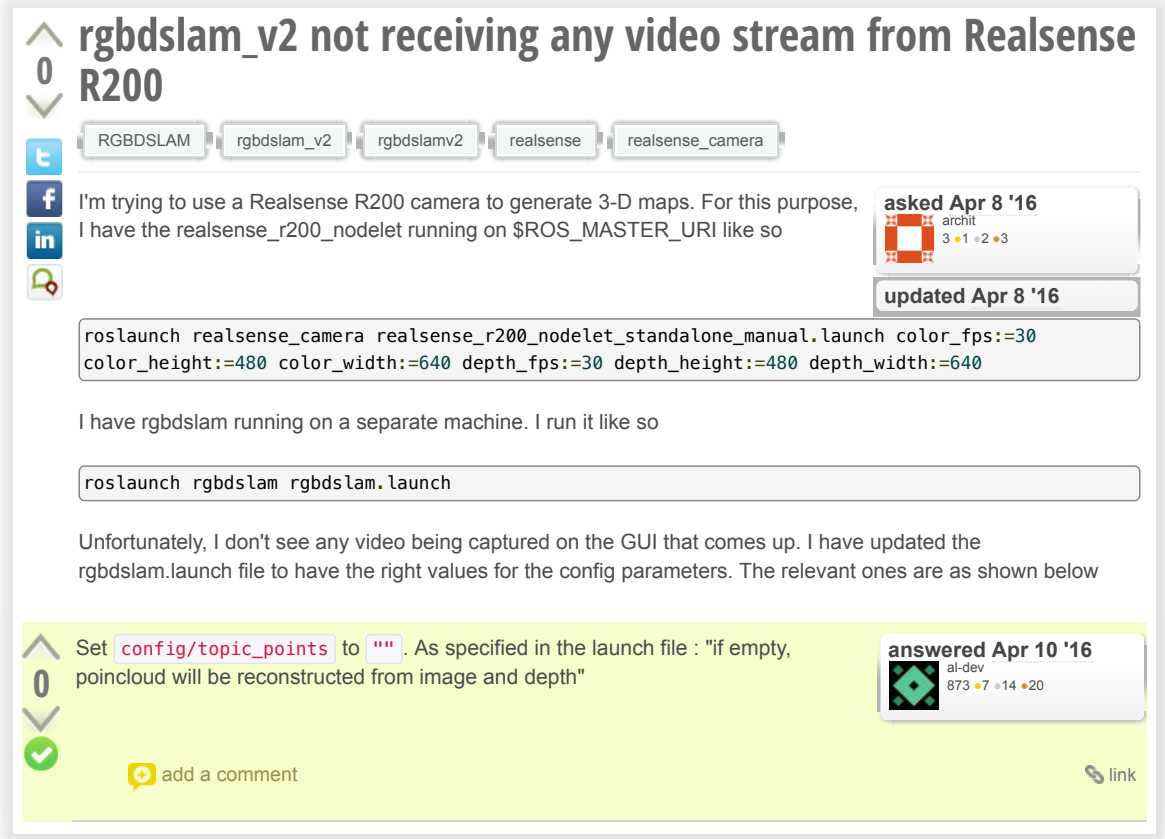
RQ2. To what extent do current techniques address these categories of misconfiguration?

Collected and filtered ROS Answers questions,



Collected and filtered ROS Answers questions,





rgbdslam_v2 not receiving any video stream from Realsense R200

0

RGBDSLAM | rgbdslam_v2 | rgbdslamv2 | realsense | realsense_camera

I'm trying to use a Realsense R200 camera to generate 3-D maps. For this purpose, I have the realsense_r200_nodelet running on $ROS_MASTER_URI like so

asked Apr 8 '16
archit
3 ●1 ●2 ●3

updated Apr 8 '16

```
roslaunch realsense_camera realsense_r200_nodelet_standalone_manual.launch color_fps:=30
color_height:=480 color_width:=640 depth_fps:=30 depth_height:=480 depth_width:=640
```

I have rgbdslam running on a separate machine. I run it like so

```
roslaunch rgbdslam rgbdslam.launch
```

Unfortunately, I don't see any video being captured on the GUI that comes up. I have updated the rgbdslam.launch file to have the right values for the config parameters. The relevant ones are as shown below

0

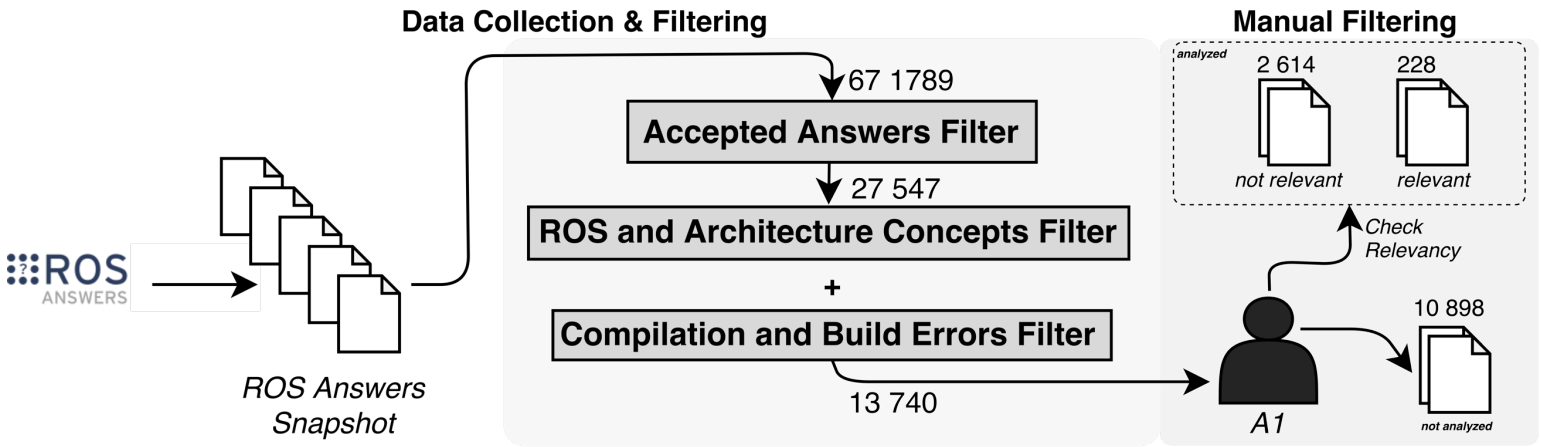
Set `config/topic_points` to `""`. As specified in the launch file : "if empty, poincloud will be reconstructed from image and depth"

answered Apr 10 '16
al-dev
873 ●7 ●14 ●20

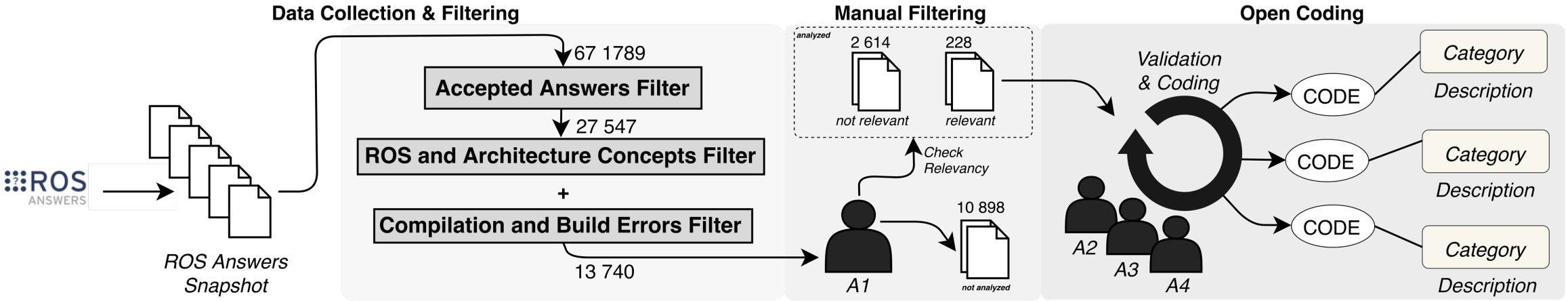
add a comment

link

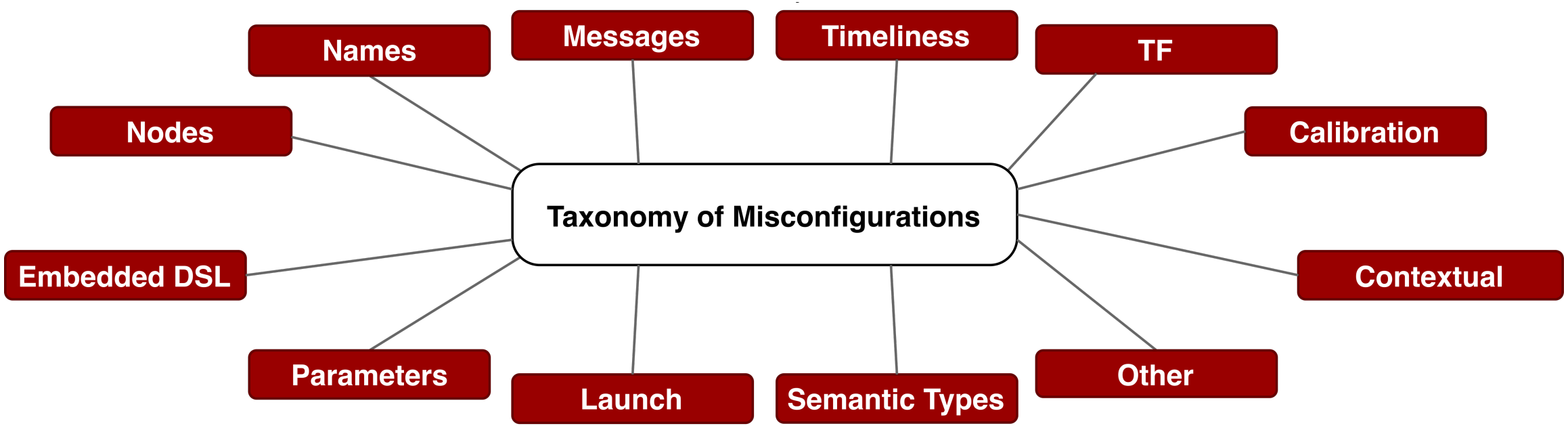
Collected and filtered ROS Answers questions, manually analyzed these,



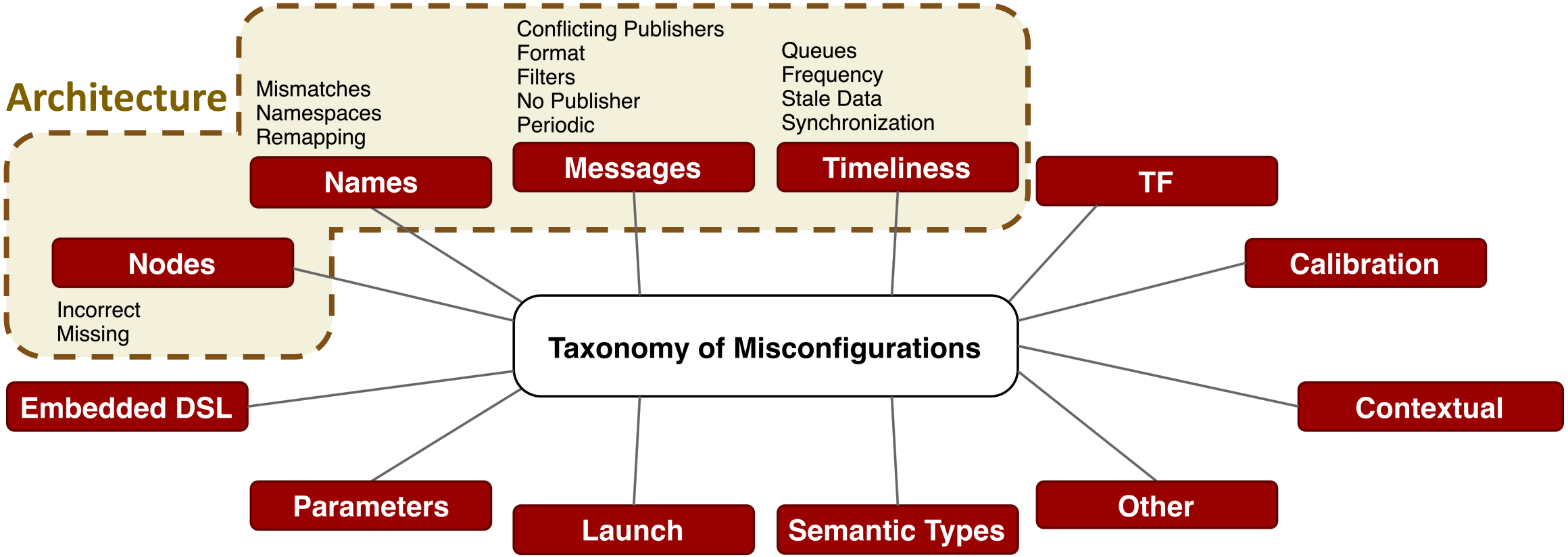
Collected and filtered ROS Answers questions, manually analyzed these, and performed open coding to obtain our taxonomy of misconfigurations



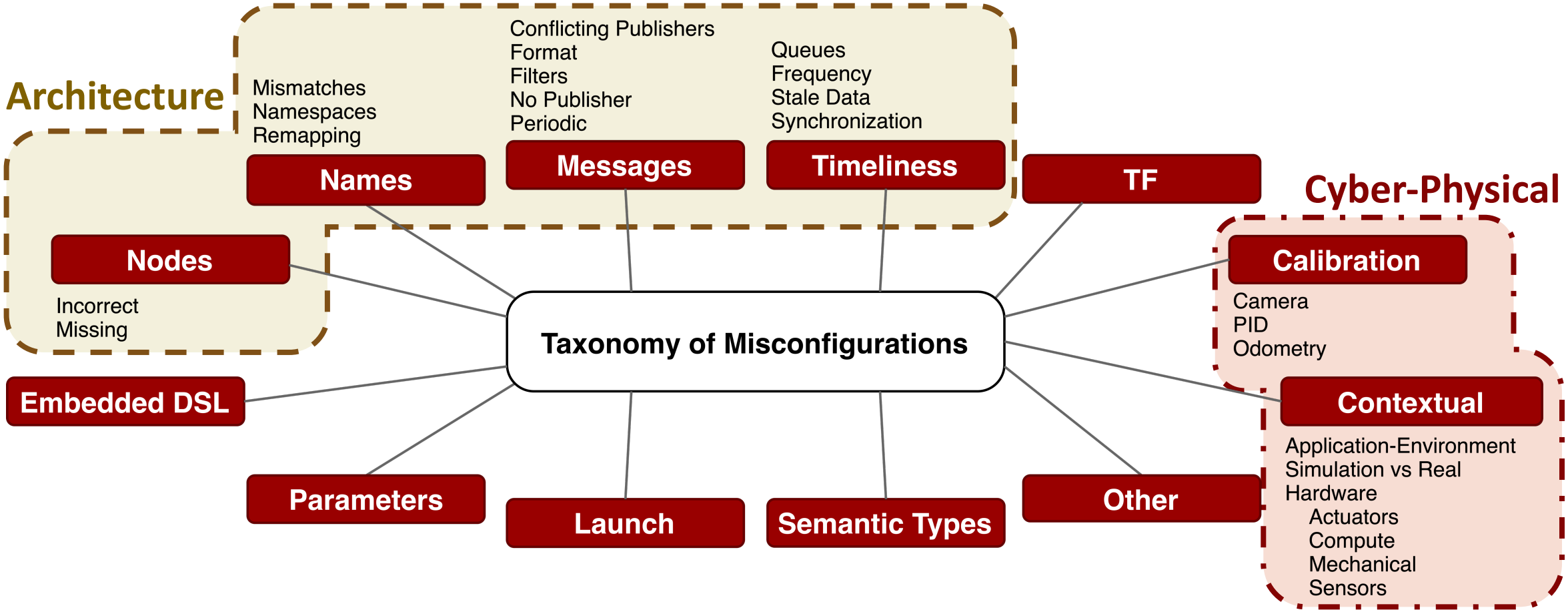
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



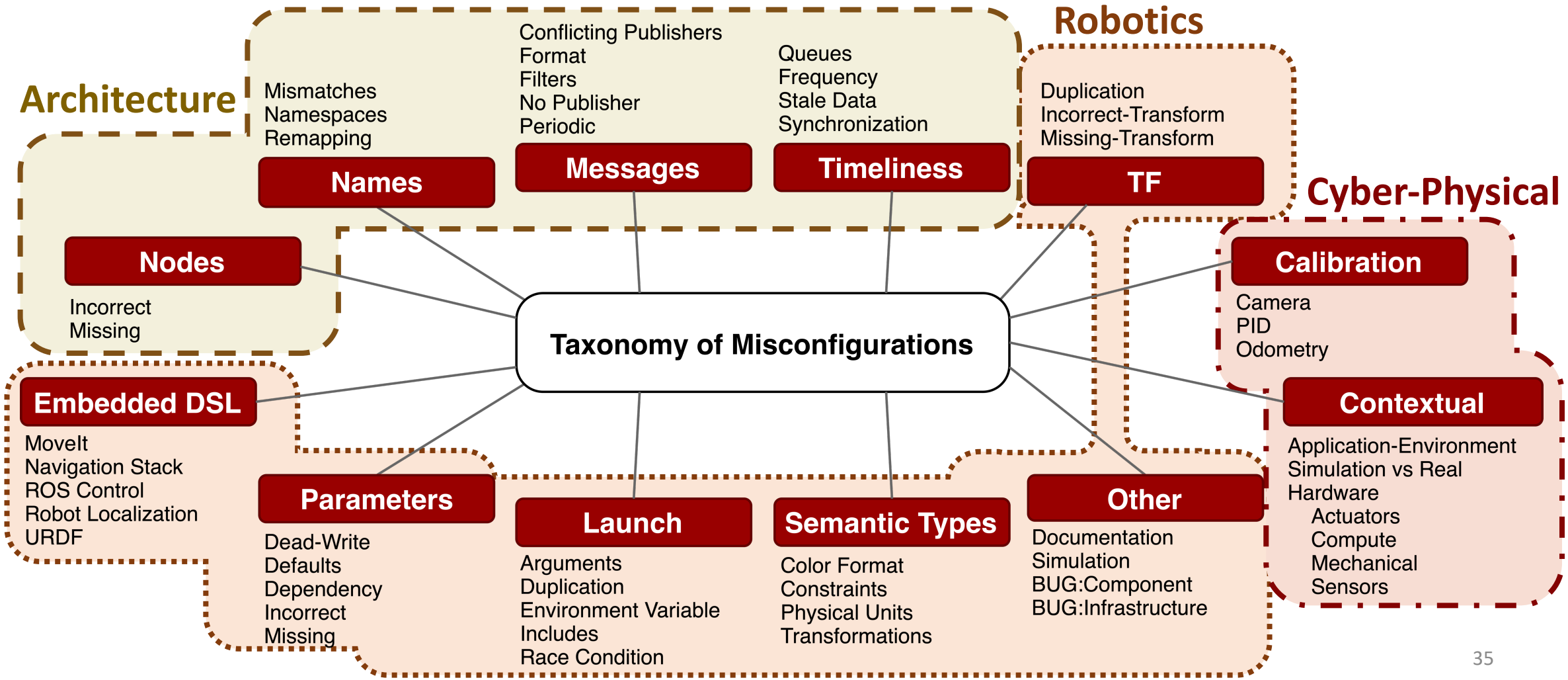
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



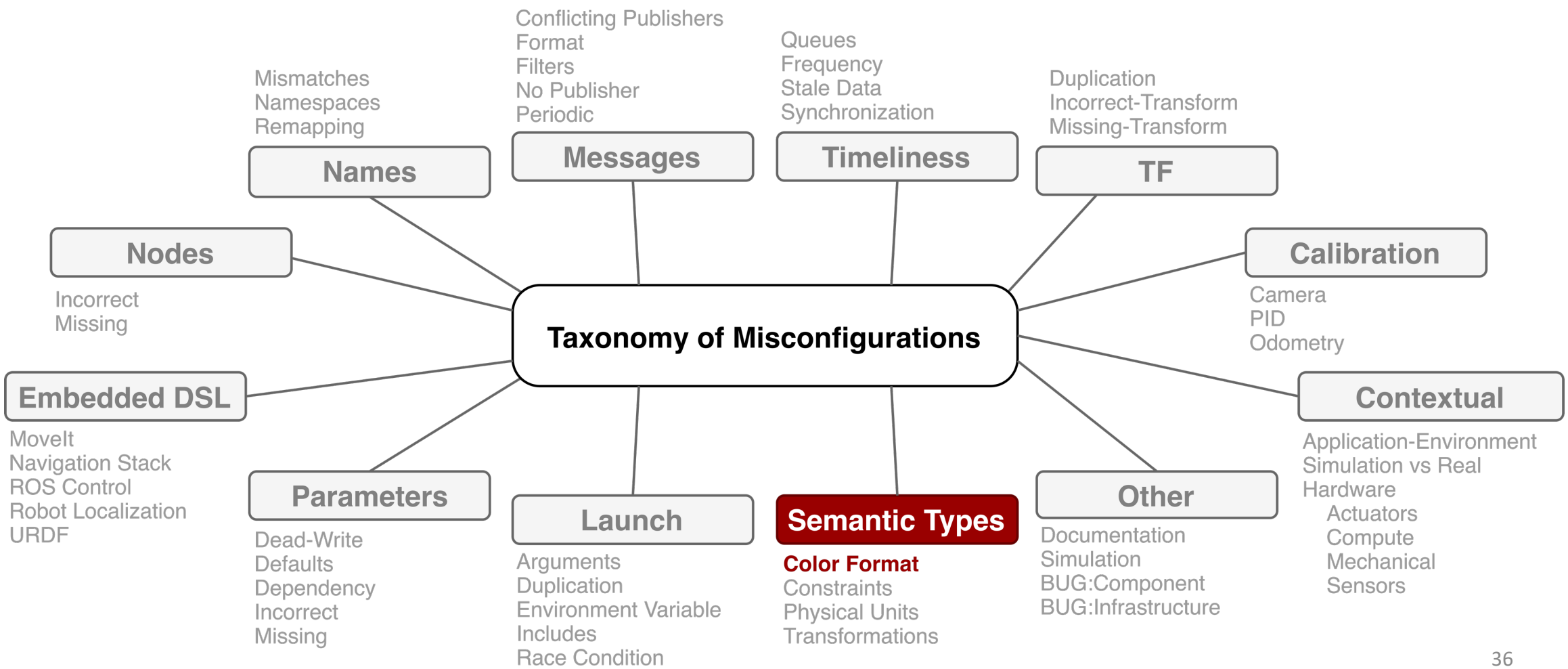
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



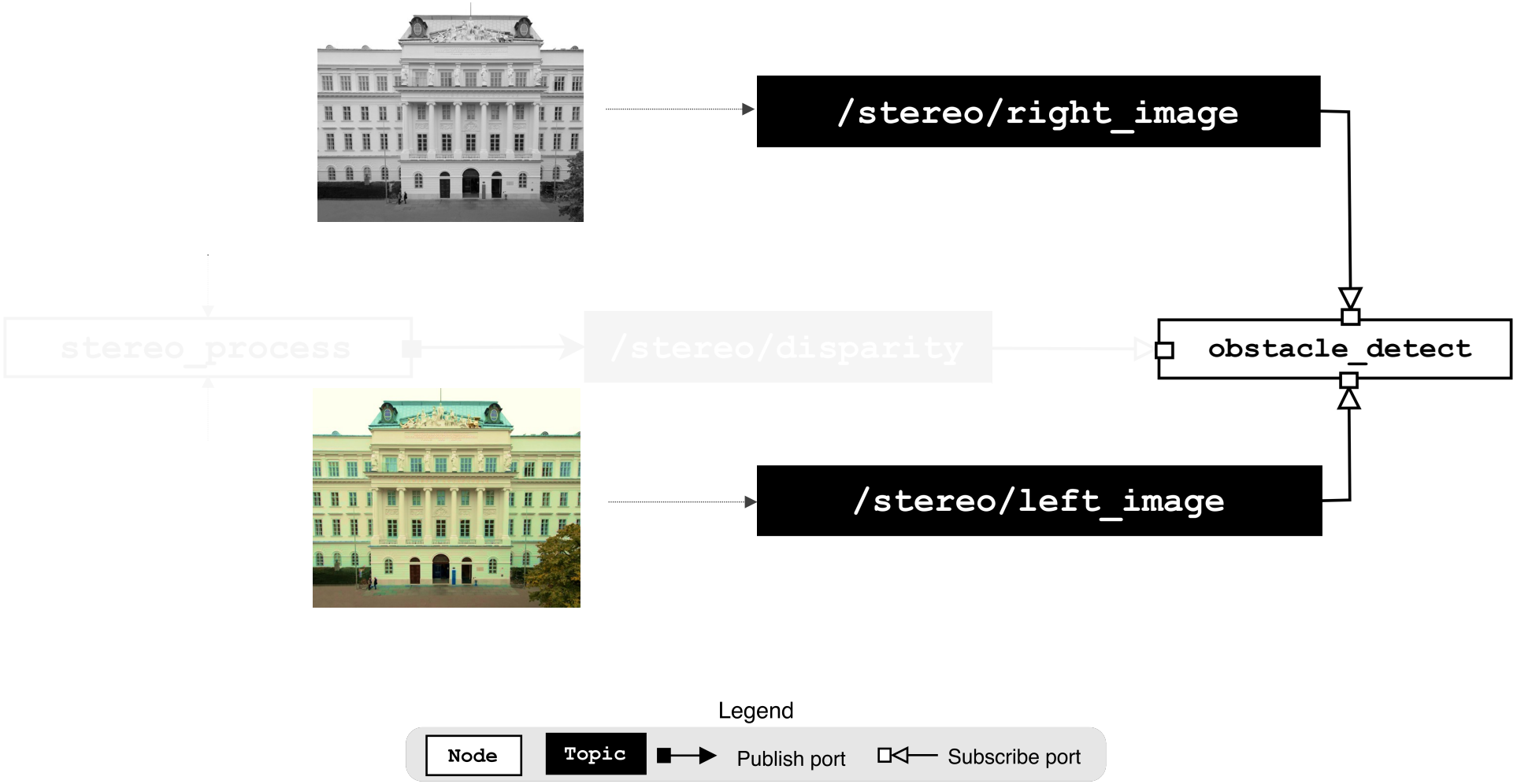
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



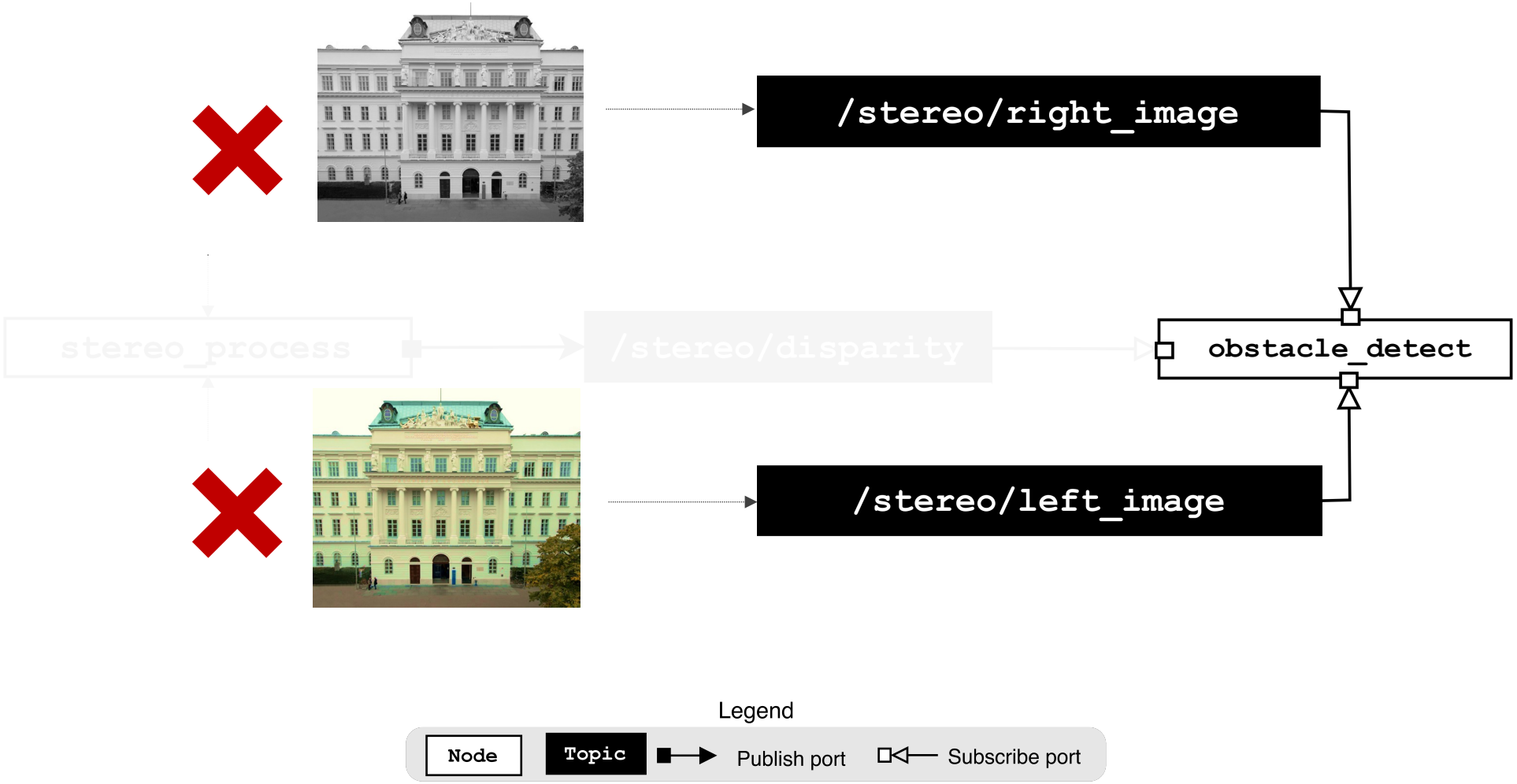
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



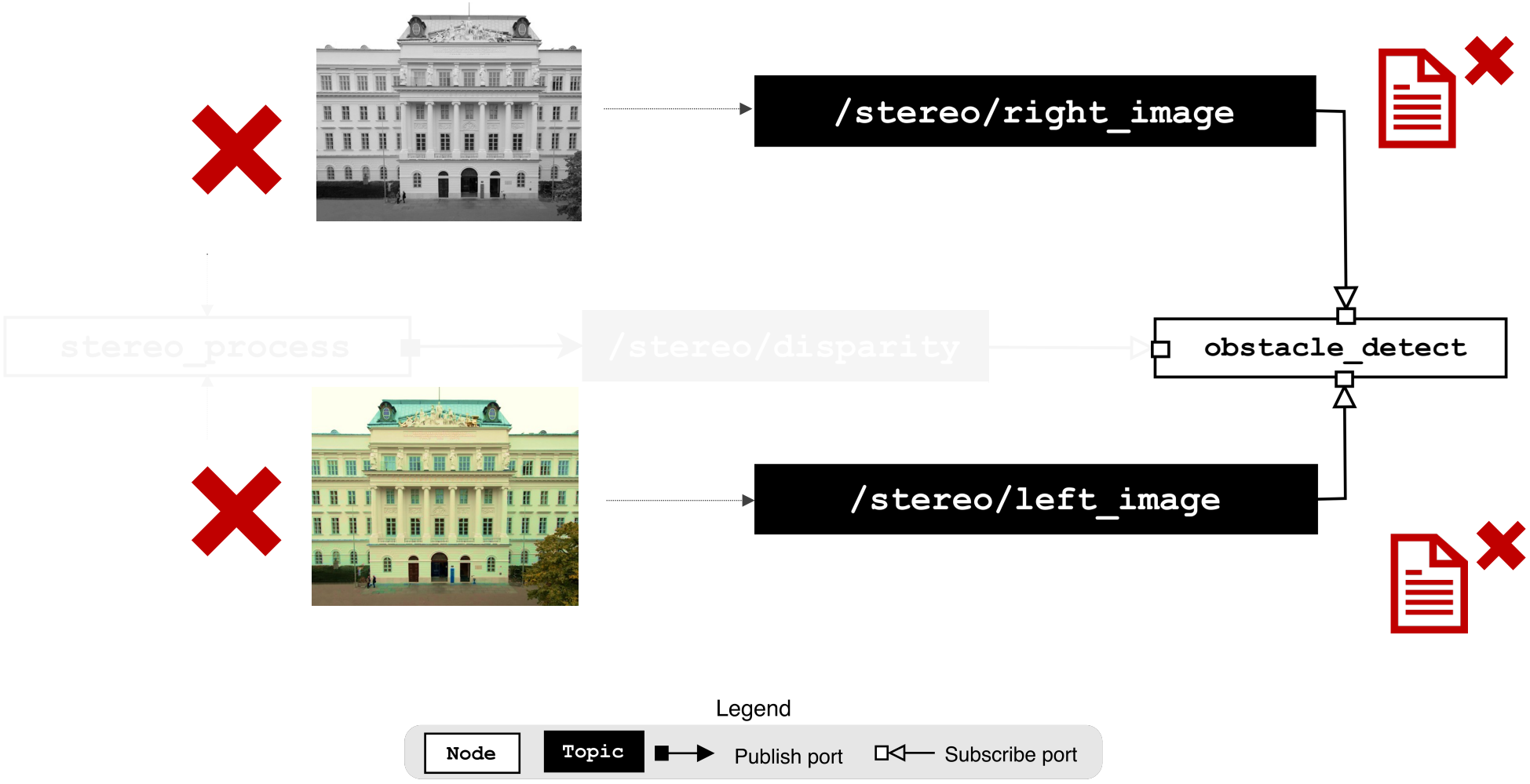
The stereo view requires both left and right images to have the same color format



The stereo view requires both left and right images to have the same color format



The stereo view requires both left and right images to have the same color format



We study the broader set of misconfigurations to identify the gap in analysis tools in detecting them

RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

RQ2. To what extent do current techniques address these categories of misconfiguration?

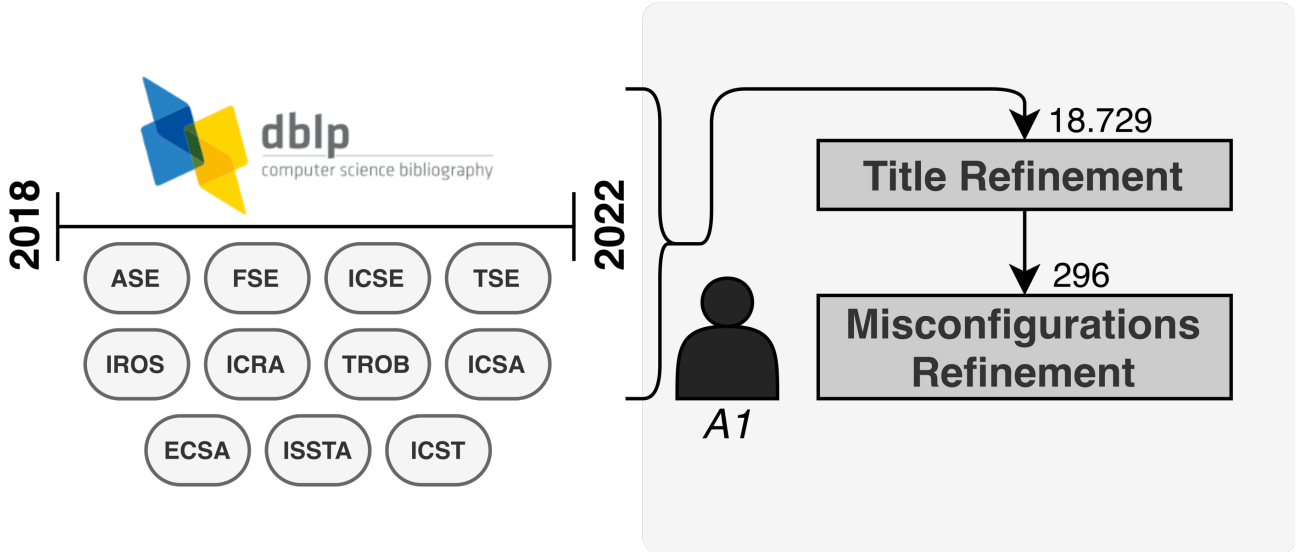
We study the broader set of misconfigurations to identify the gap in analysis tools in detecting them

RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

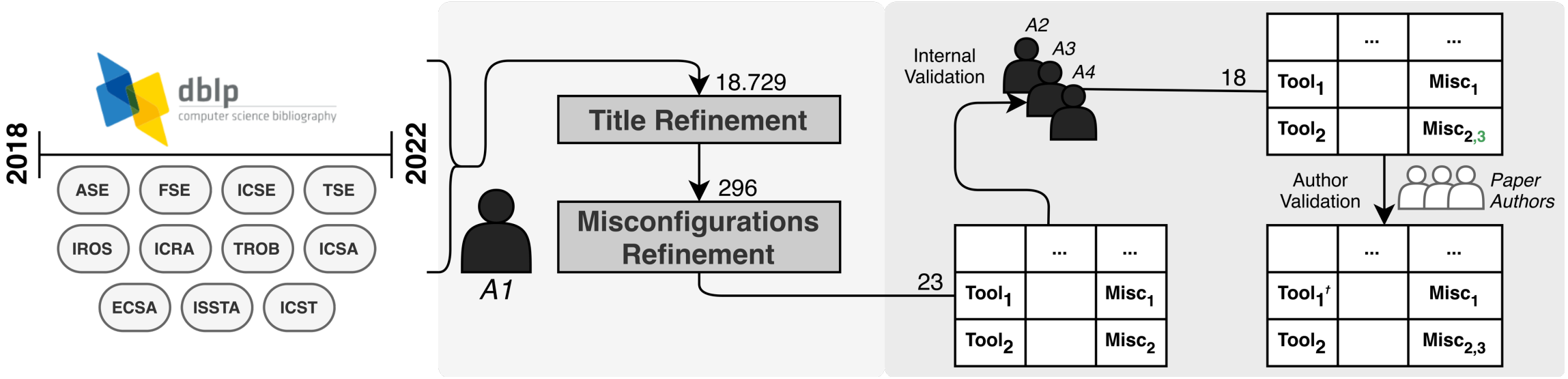
RQ2. To what extent do current techniques address these categories of misconfiguration?

Collected prior work from 11 top venues in software engineering, testing, architecture and robotics

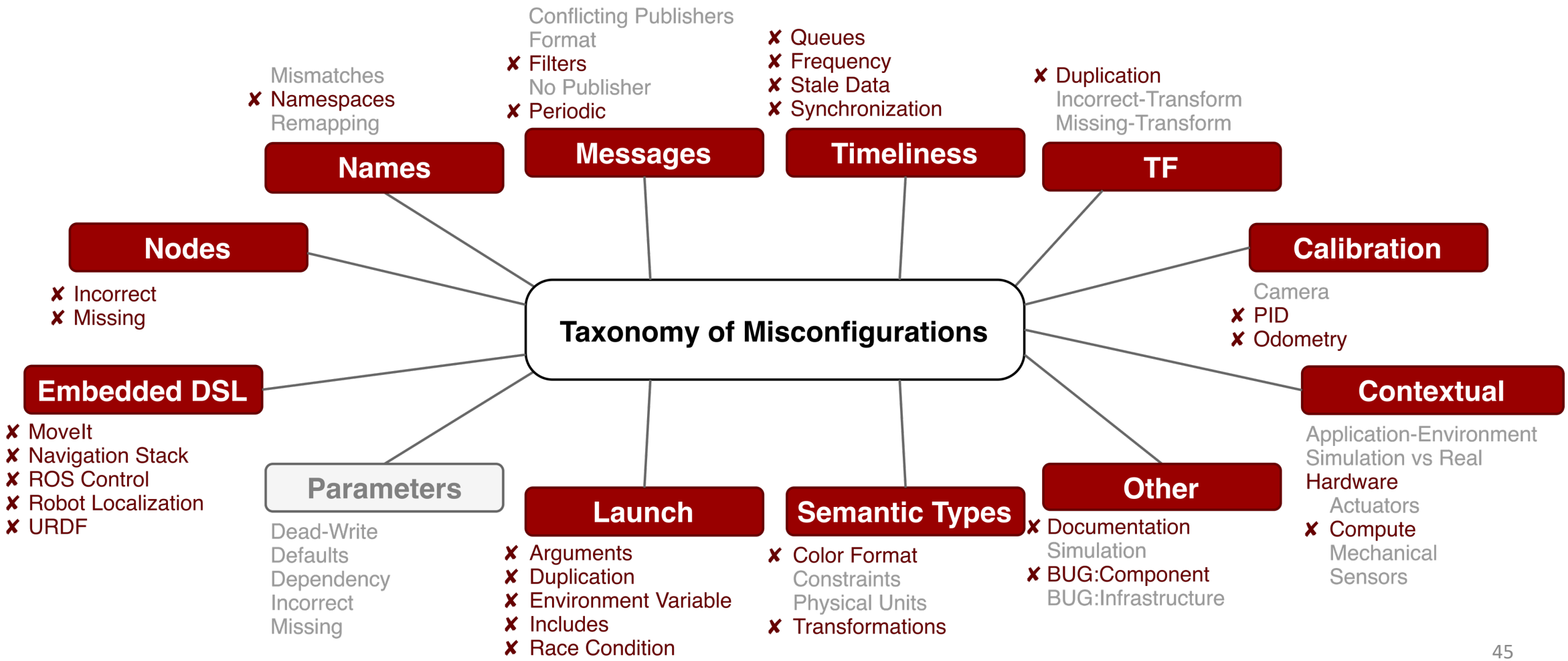
Manually refined each paper according to its ability to address any of the detect misconfigurations



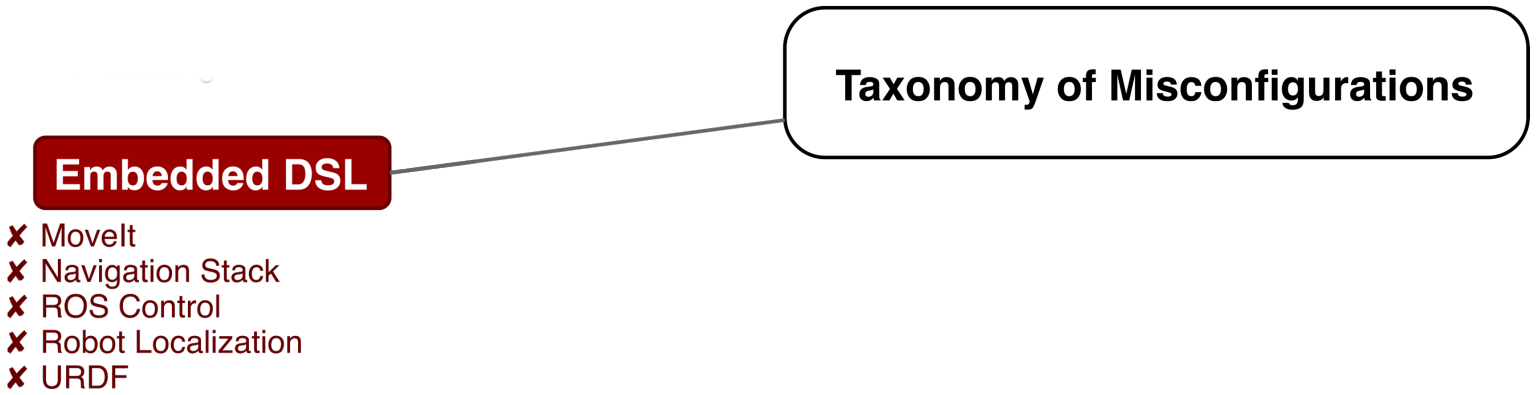
Performed internal validation to reduce biases, and external validation to confirm our assumptions



27 of 50 categories of misconfigurations are not addressed by current techniques



Domain-Specific Languages (DSL) are critical for defining ROS configurations → Analysis tools must analyze these



Domain-Specific Languages (DSL) are critical for defining ROS configurations → Analysis tools must analyze these

Taxonomy of Misconfigurations

Embedded DSL

- ✘ MoveIt
- ✘ Navigation Stack
- ✘ ROS Control
- ✘ Robot Localization
- ✘ URDF

```
TrajectoryPlanner:
  max_vel_x: 0.5
  min_vel_x: 0.1
  max_vel_theta: 1.0
  min_vel_theta: 0.2
  acc_lim_x: 0.2
  acc_lim_y: 0.2
  acc_lim_theta: -2.0
  holonomic_robot: false
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 0.2
  pdist_scale: 0.6
  gdist_scale: 0.8
  occdist_scale: 0.01
  max_trans_vel: 0.6
  min_trans_vel: 0.1
  recovery_behavior_enabled:

  # Obstacle avoidance
  sim_time: 1.5
  vx_samples: 20
```

Domain-Specific Languages (DSL) are critical for defining ROS configurations → Analysis tools must analyze these

Taxonomy of Misconfigurations

Embedded DSL

- ✘ MoveIt
- ✘ Navigation Stack
- ✘ ROS Control
- ✘ Robot Localization
- ✘ URDF

```
TrajectoryPlanner:
  max_vel_x: 0.5
  min_vel_x: 0.1
  max_vel_theta: 1.0
  min_vel_theta: 0.2
  acc_lim_x: 0.2
  acc_lim_y: 0.2
  acc_lim_theta: -2.0
  holonomic_robot: false
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 0.2
  pdist_scale: 0.6
  gdist_scale: 0.8
  occdist_scale: 0.01
  max_trans_vel: 0.6
  min_trans_vel: 0.1
  recovery_behavior_enabled:

  # Obstacle avoidance
  sim_time: 1.5
  vx_samples: 20
```

Domain-Specific Languages (DSL) are critical for defining ROS configurations → Analysis tools must analyze these

Taxonomy of Misconfigurations

Embedded DSL

- ✘ MoveIt
- ✘ Navigation Stack
- ✘ ROS Control
- ✘ Robot Localization
- ✘ URDF

Documentation is not enforced

```
TrajectoryPlanner:
  max_vel_x: 0.5
  min_vel_x: 0.1
  max_vel_theta: 1.0
  min_vel_theta: 0.2
  acc_lim_x: 0.2
  acc_lim_y: 0.2
  acc_lim_theta: -2.0
  holonomic_robot: false
  yaw_goal_tolerance: 0.1
  xy_goal_tolerance: 0.2
  pdist_scale: 0.6
  gdist_scale: 0.8
  occdist_scale: 0.01
  max_trans_vel: 0.6
  min_trans_vel: 0.1
  recovery_behavior_enabled:

  # Obstacle avoidance
  sim_time: 1.5
  vx_samples: 20
```

We identify three future research directions to detect misconfigurations in ROS systems. Analysis Tools must…

1. … work with ROS domain-specific language and dialects;

2. … consider information about the robot's hardware, physical environment and intended application;

3. … analyze run-time behavior because static analysis is not sufficient to detect all misconfigurations.

We identify three future research directions to detect misconfigurations in ROS systems. Analysis Tools must…

1. … work with ROS domain-specific language and dialects;

2. … consider information about the robot's hardware, physical environment and intended application;

3. … analyze run-time behavior because static analysis is not sufficient to detect all misconfigurations.

We identify three future research directions to detect misconfigurations in ROS systems. Analysis Tools must…

1. … work with ROS domain-specific language and dialects;

2. … consider information about the robot's hardware, physical environment and intended application;

3. … analyze run-time behavior because static analysis is not sufficient to detect all misconfigurations.

Understanding Misconfigurations in ROS: An Empirical Study and Current Approaches

Paulo Canelas
with Bradley Schmerl, Alcides Fonseca, and Christopher S. Timperley
Carnegie Mellon University
University of Lisbon

International Symposium on Software Testing and Analysis (ISSTA). 2024.

DATASET OF MISCONFIGURATIONS

Robot software has a lot of challenges and sources of misconfigurations yet to be addressed!

https://pcanelas.com