

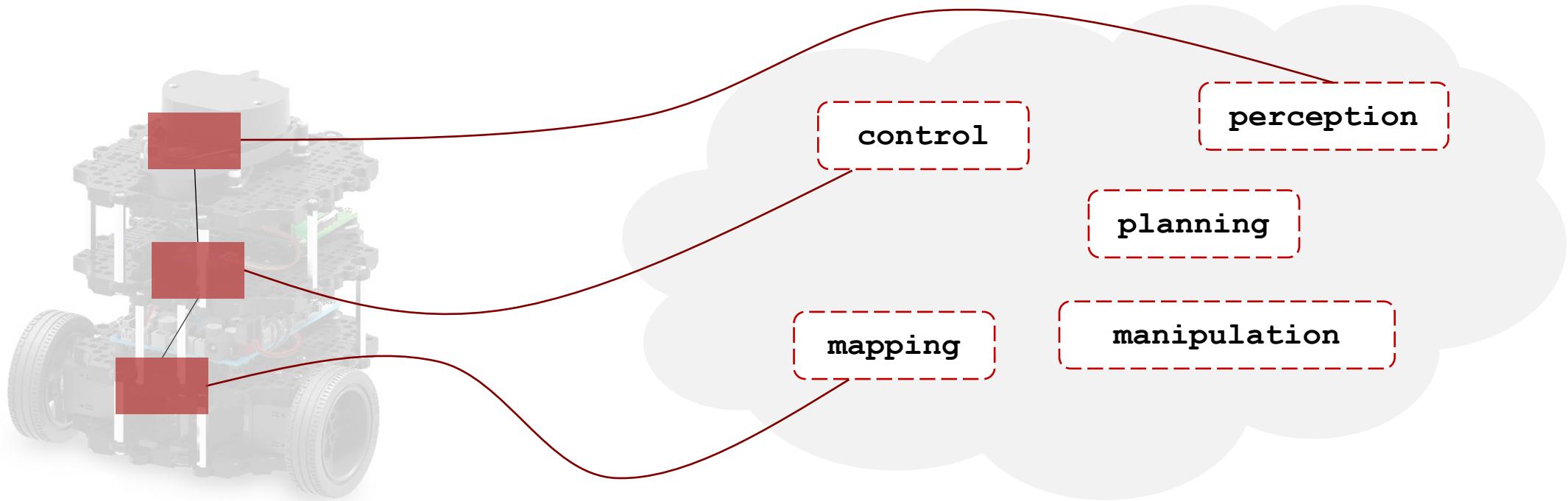
Understanding Misconfigurations in ROS: An Empirical Study and Current Approaches

Paulo Canelas

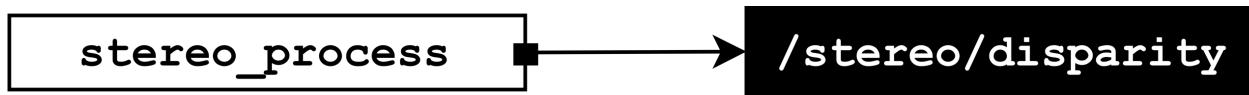
with Bradley Schmerl, Alcides Fonseca, and Christopher S. Timperley
Carnegie Mellon University
University of Lisbon

First Draft of the Presentation for International Symposium on Software Testing and Analysis (ISSTA). 2024.

The Robot Operating System (ROS) allows developers to integrate reusable, off-the-shelf components



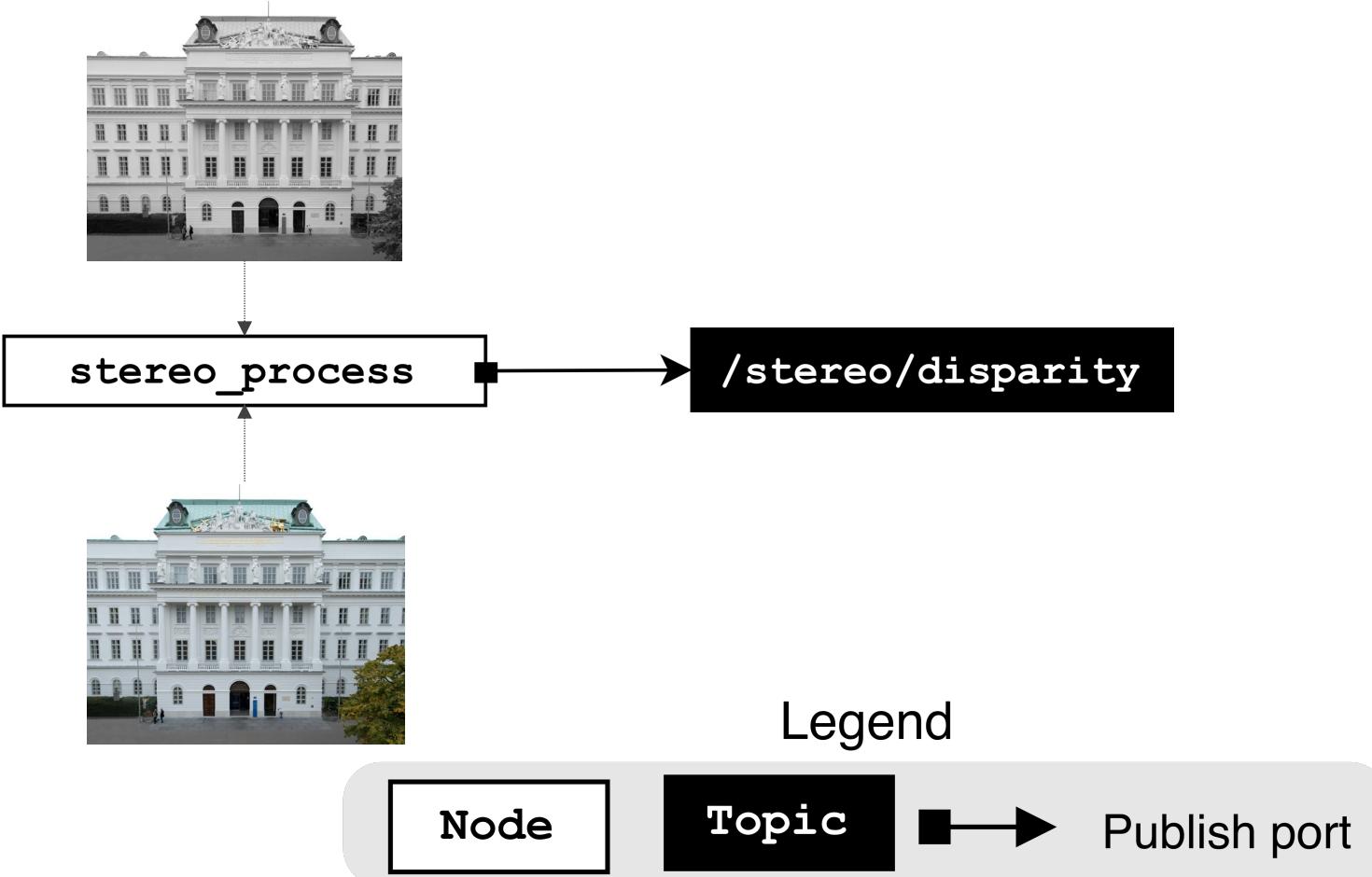
Components are processes that process inputs and may produce an output (e.g., nodes in publisher-subscriber)



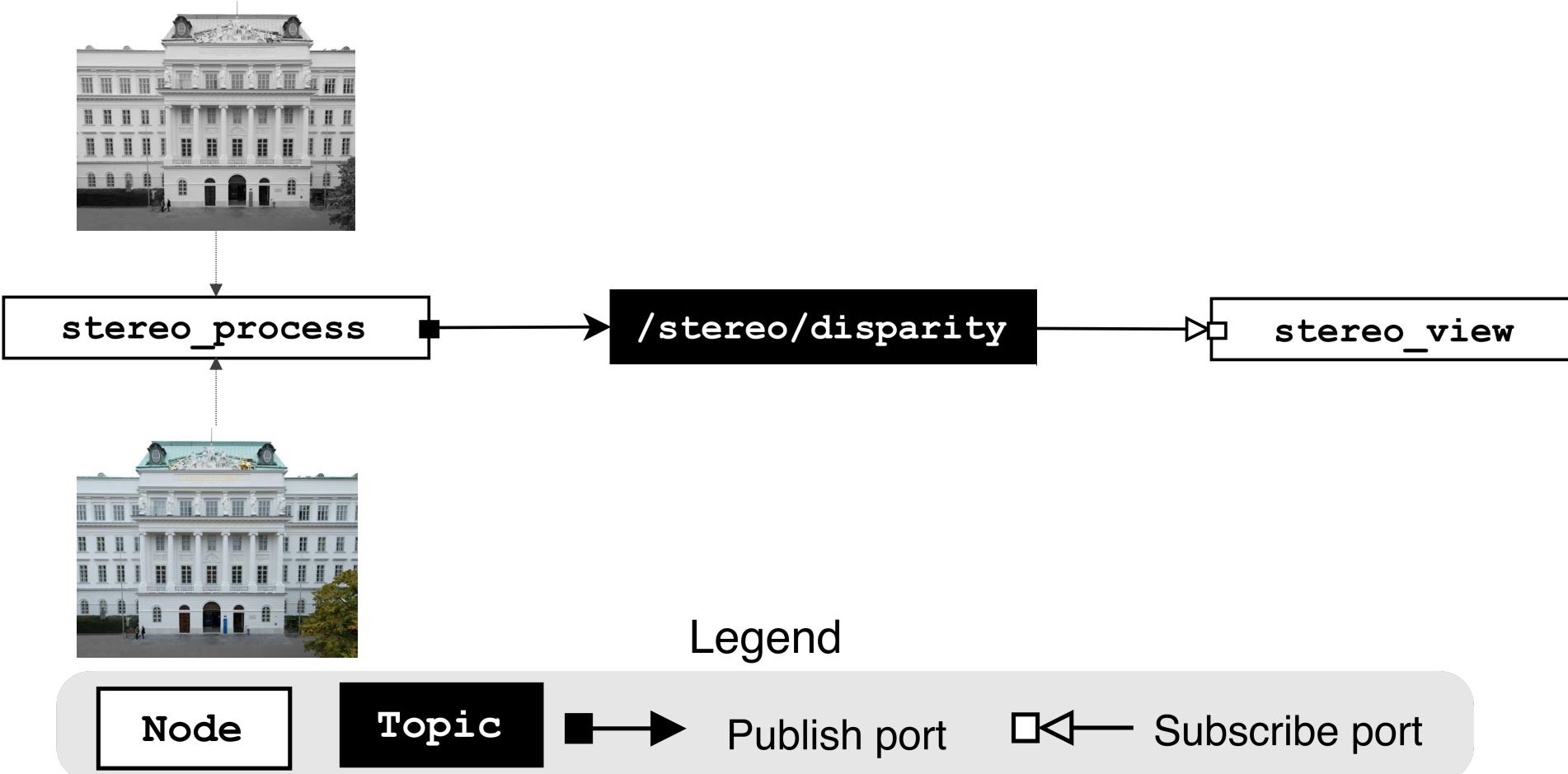
Legend



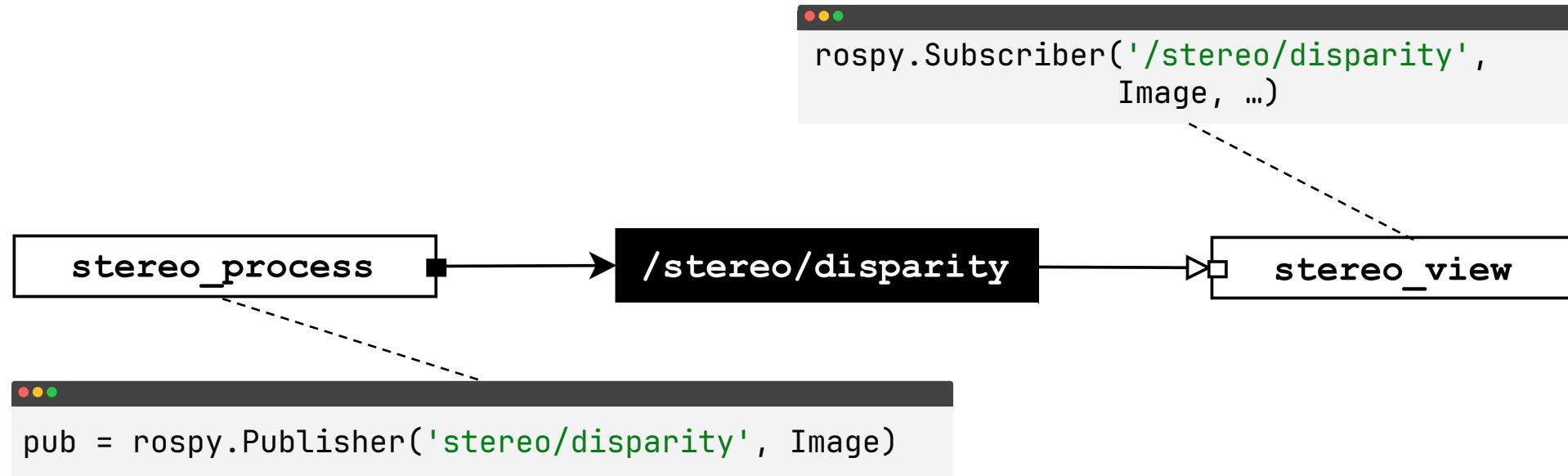
Components are processes that process inputs and may produce an output (e.g., nodes in publisher-subscriber)



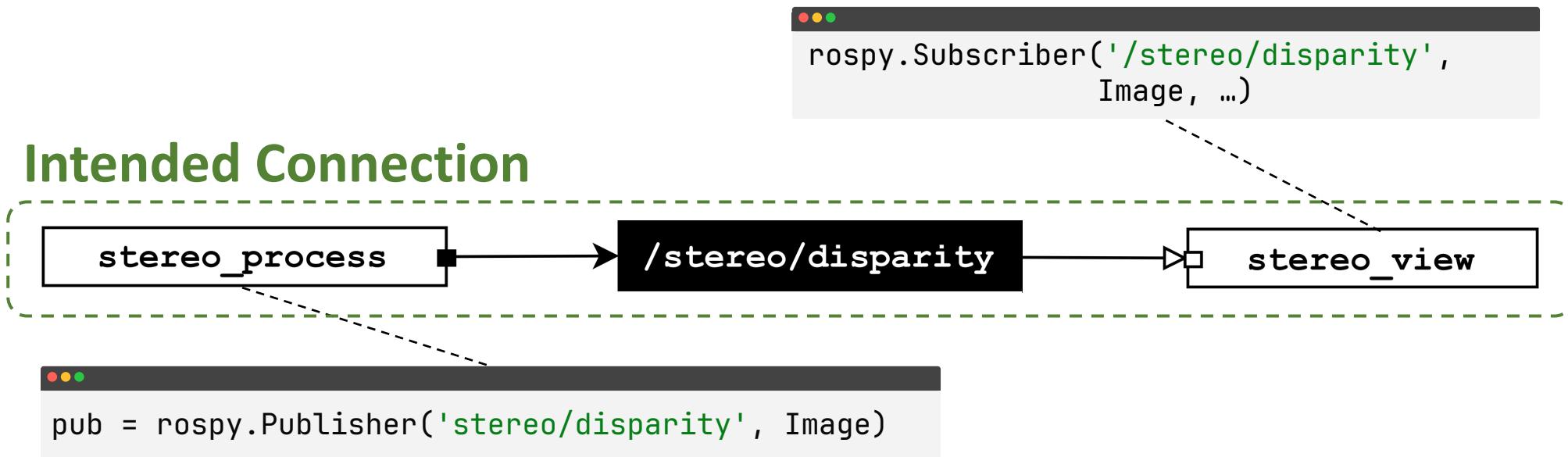
Components are processes that process inputs and may produce an output (e.g., nodes in publisher-subscriber)



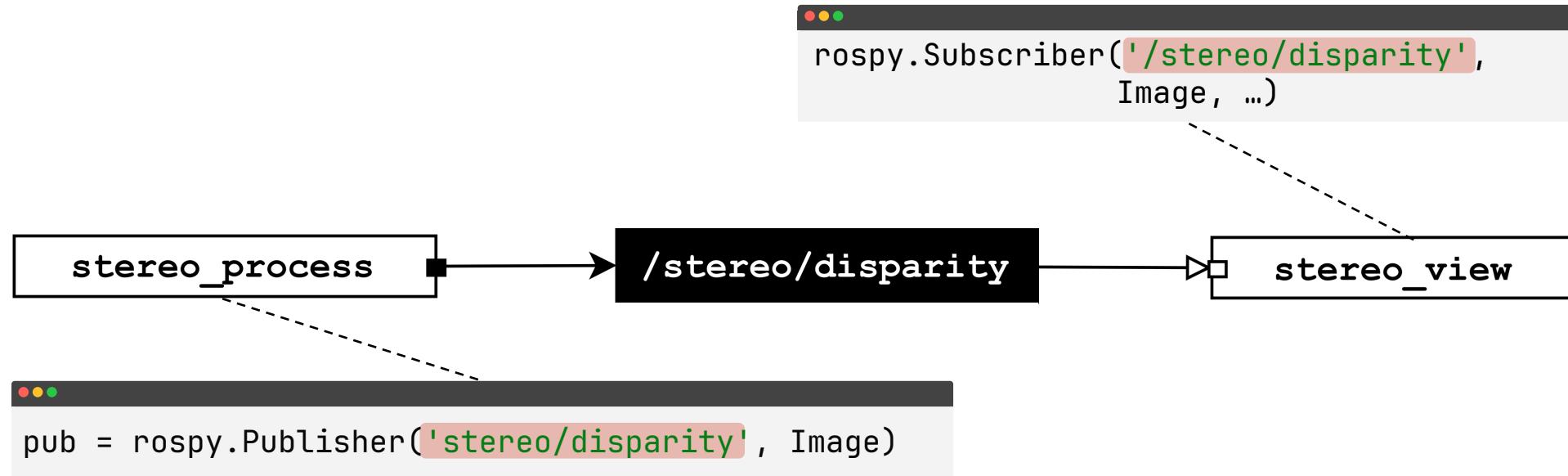
It is up to developers to guarantee that components assumptions match and the system is well configured



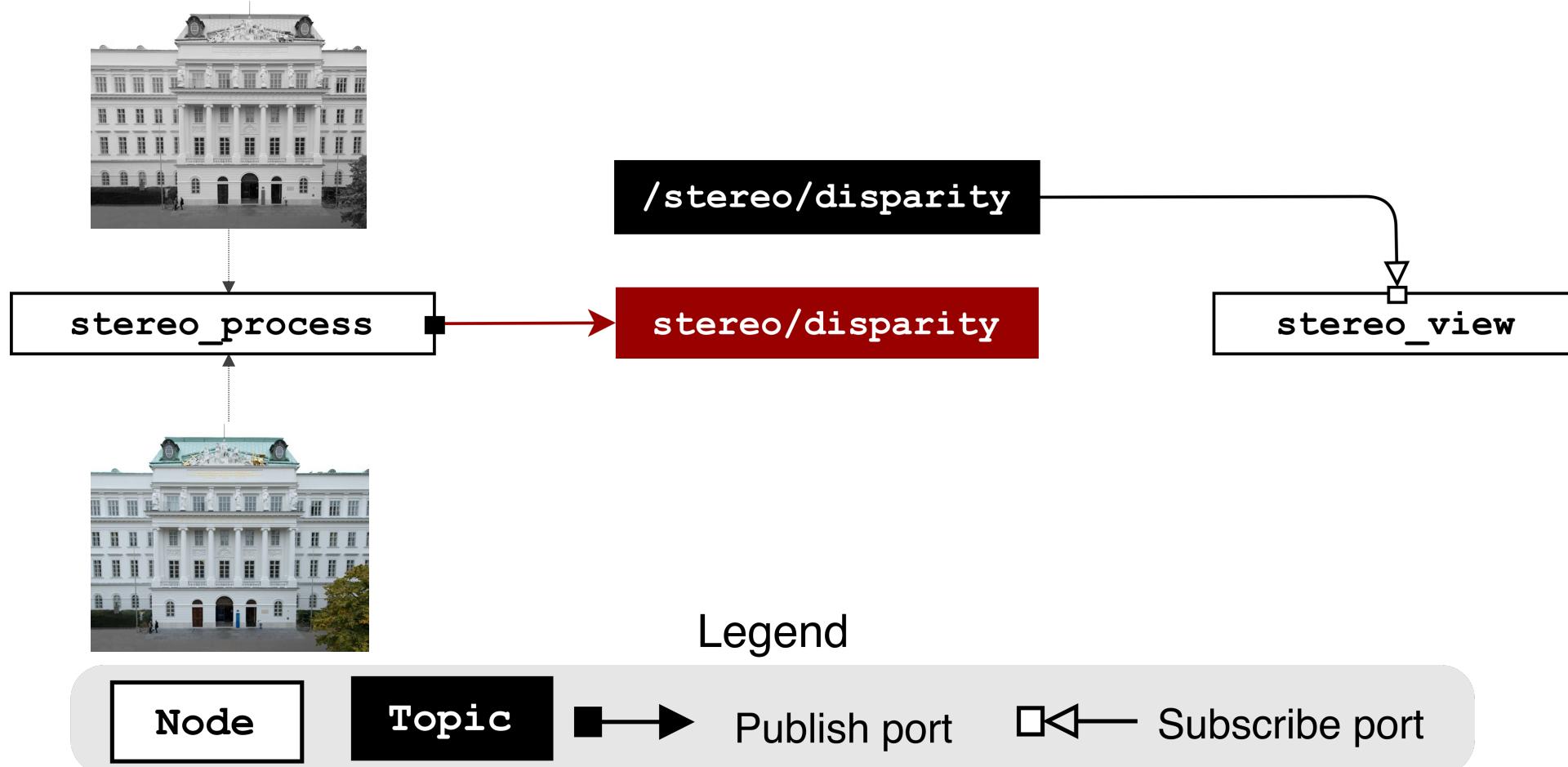
It is up to developers to guarantee that components assumptions match and the system is well configured



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



Misconfigurations arise while integrating components due to mismatched assumptions



We must understand misconfigurations in ROS systems to be able to detect and prevent them

─ **HAROS** [Santos et al, 2021] and **ROSDiscover** [Timperley et al, 2022]

→ **Structural Architectural Misconfigurations**

─ **PHYSFRAME** [Kate et al, 2021]

→ **Frame Coordinates Errors**

─ **Phys** [Kate et al, 2018]

→ **Physical Unit Mismatches**

In this work, we study what is the broader set of misconfigurations and which tools detect them

In this work, we study what is the broader set of misconfigurations and which tools detect them

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

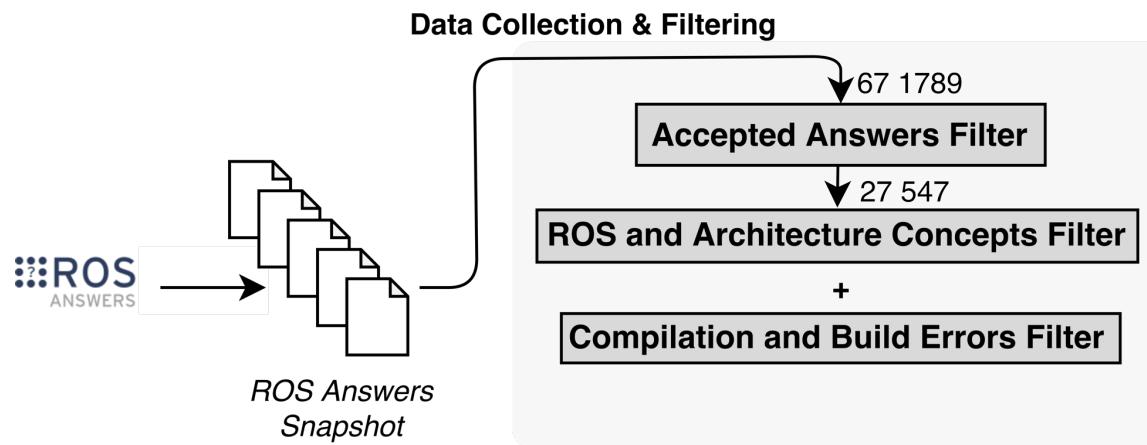
RQ2. To what extent do current techniques address these categories of misconfiguration?

In this work, we study what is the broader set of misconfigurations and which tools detect them

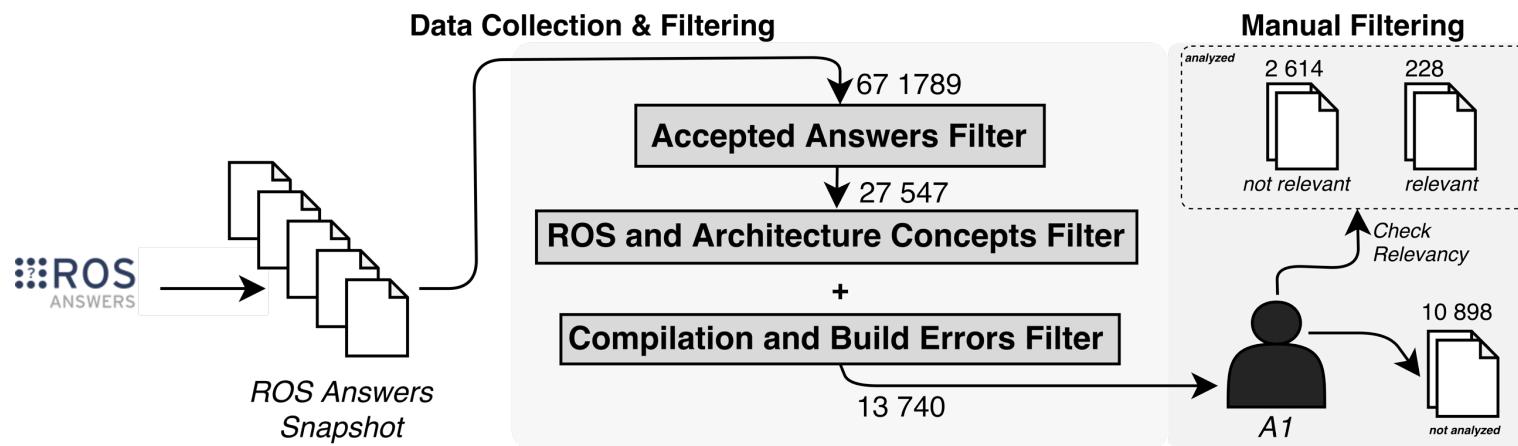
RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

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

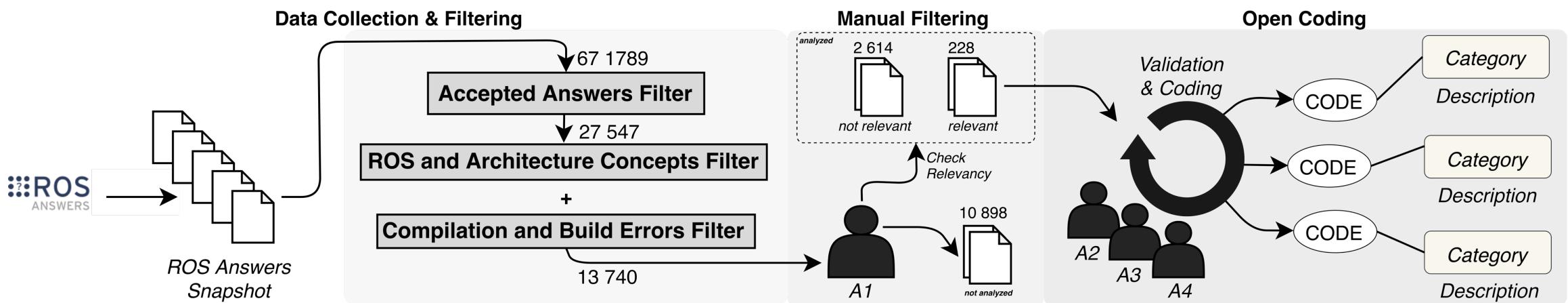
Collected and filtered ROS Answers questions,



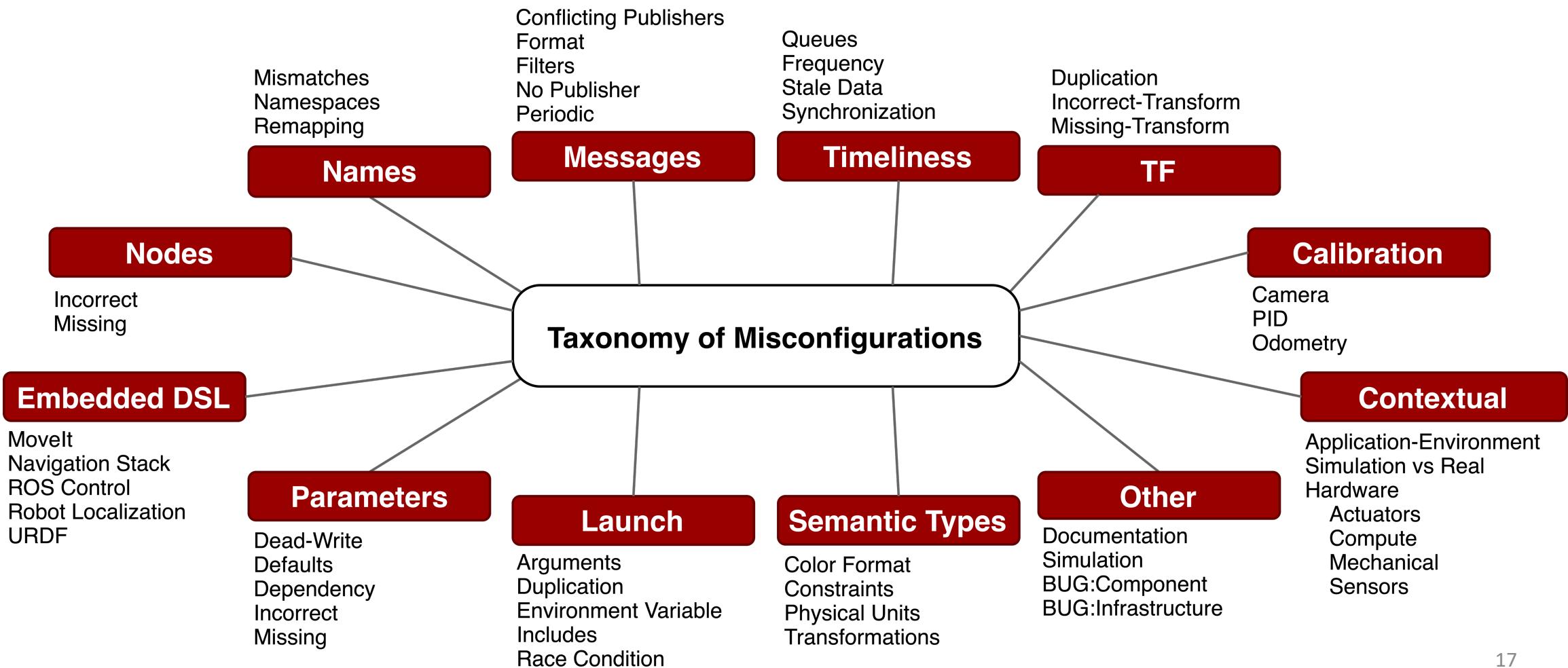
Collected and filtered ROS Answers questions, manually analyzed these,



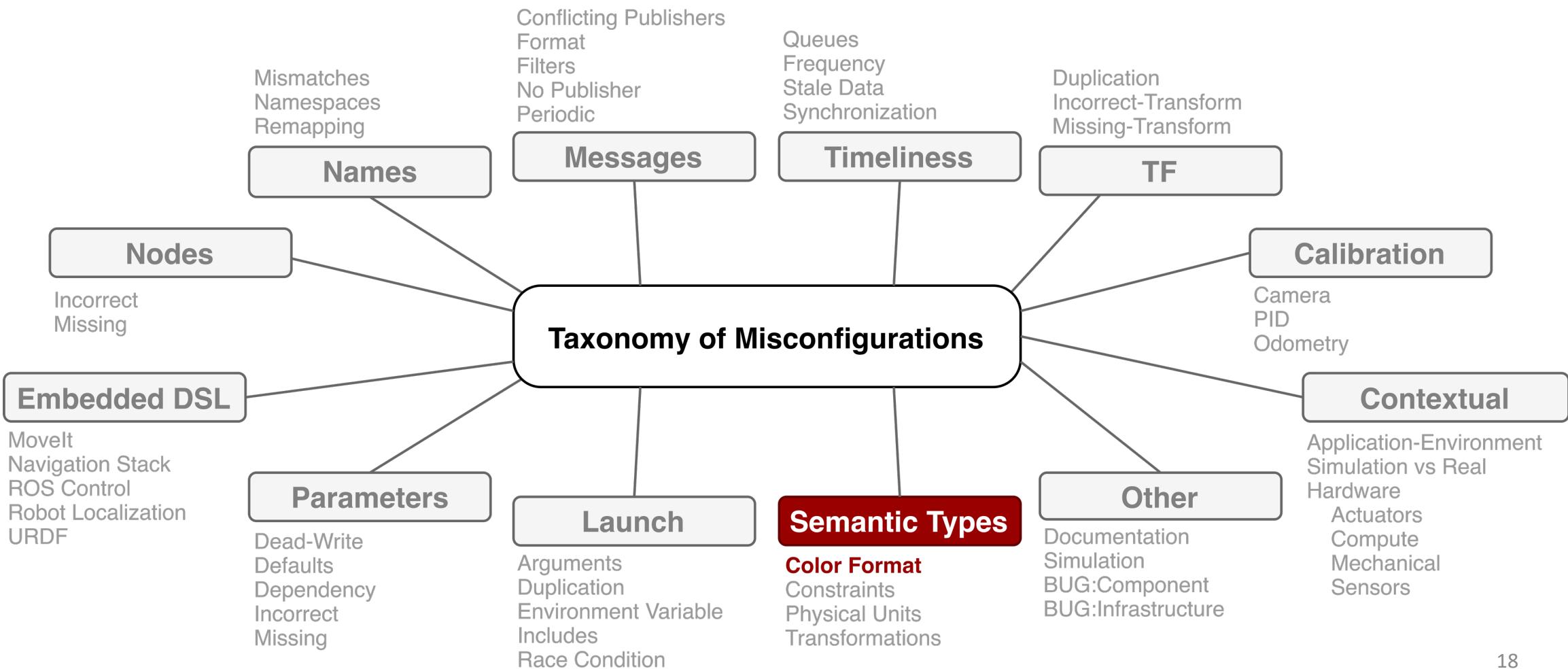
Collected and filtered ROS Answers questions, manually analyzed these, and performed open coding to obtain our taxonomy of misconfigurations



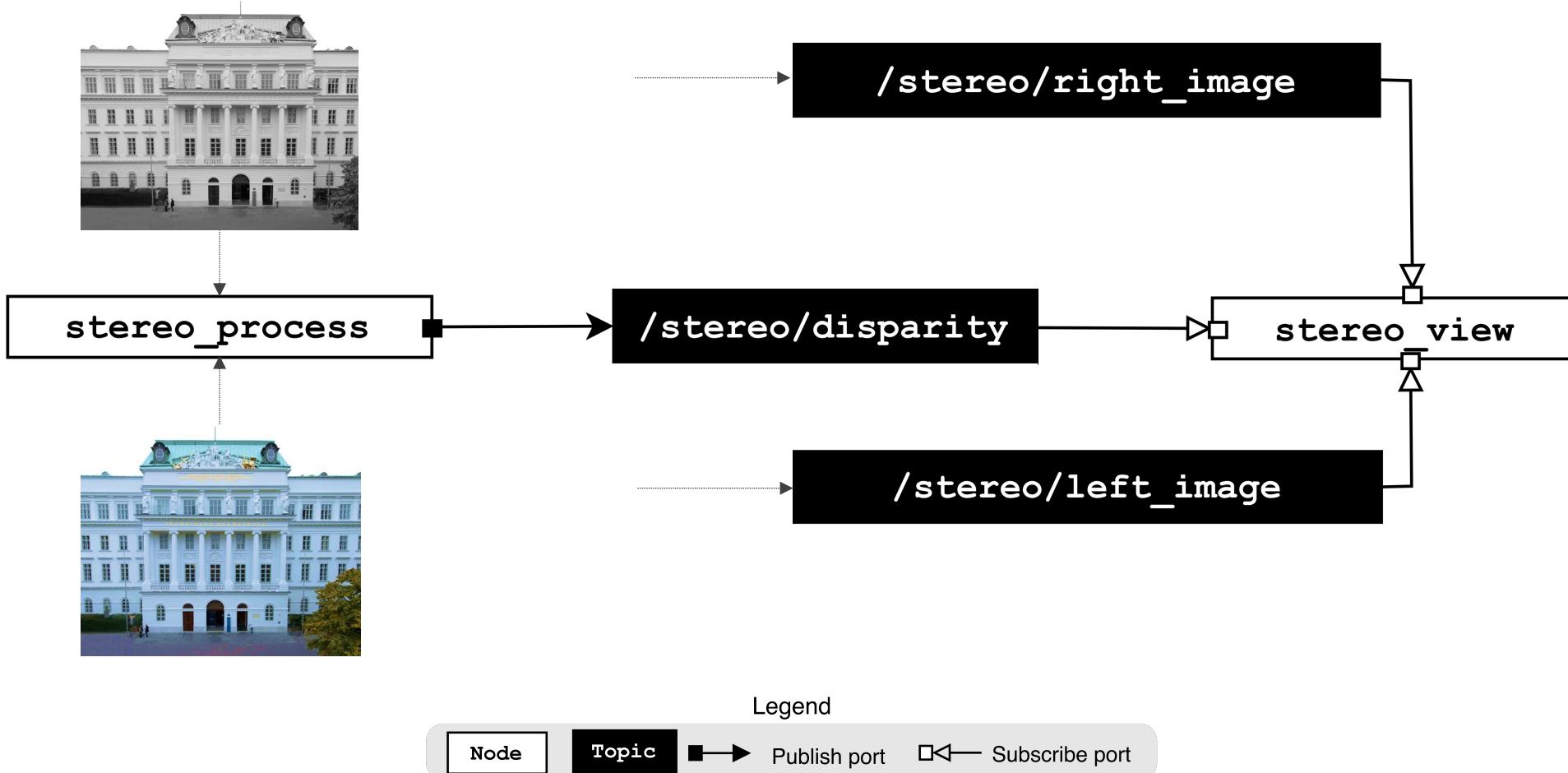
Obtained a taxonomy of 12 high-level categories and 50 subcategories of misconfigurations



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



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



In this work, we study what is the broader set of misconfigurations and which tools detect 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?

In this work, we study what is the broader set of misconfigurations and which tools detect them

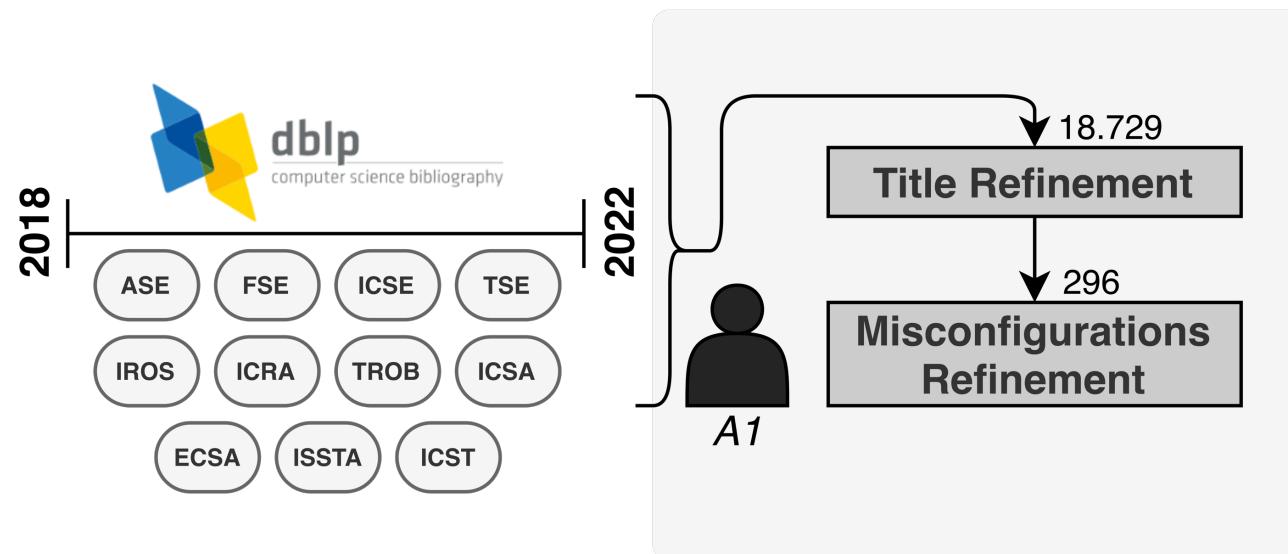
RQ1. What kinds of misconfigurations do developers make when building robot software systems with ROS?

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

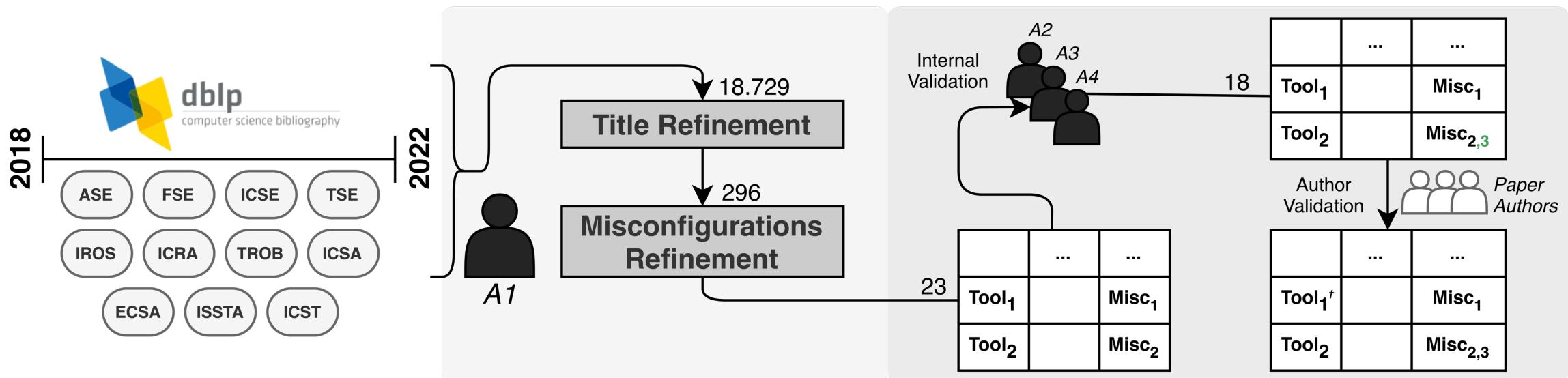
Collected prior work from 11 top venues in software engineering, testing, architecture and robotics



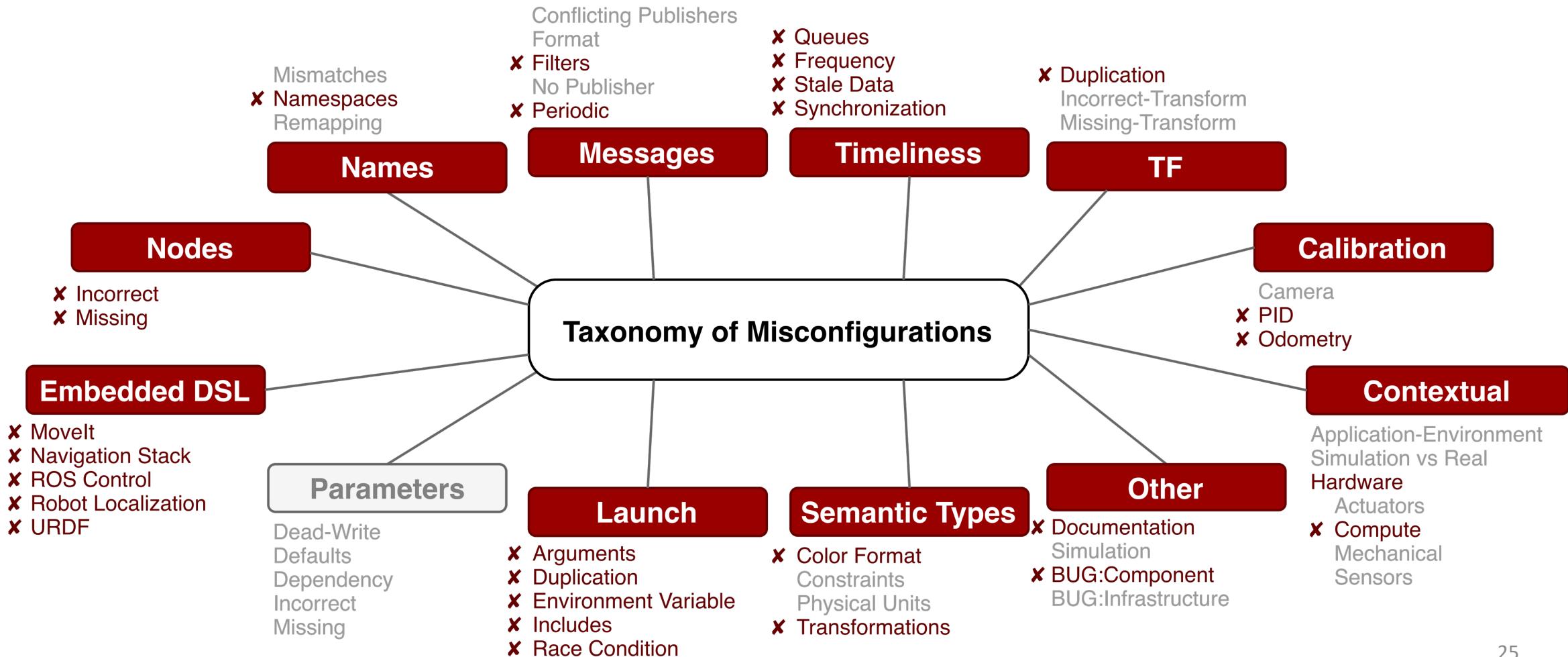
Manually refined each paper according to its ability to address any of the detect misconfigurations



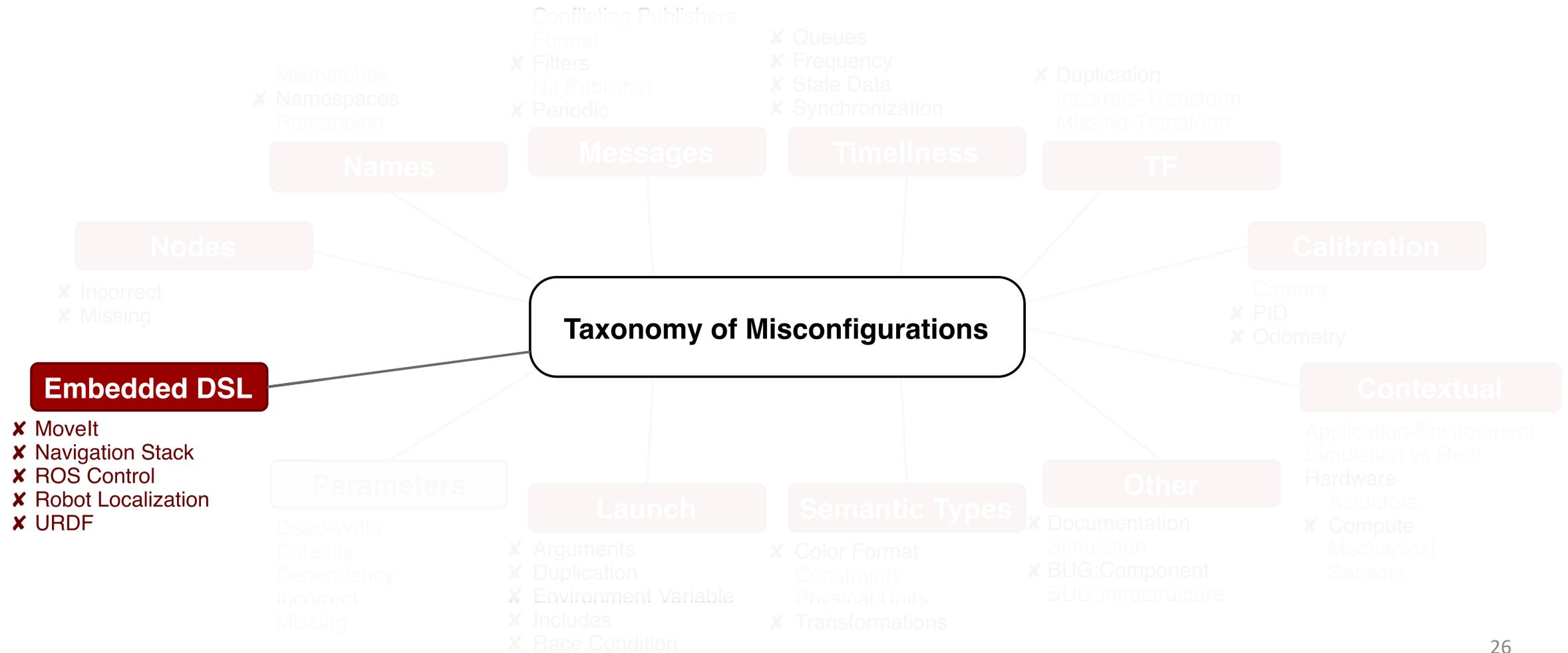
Performed internal validation to reduce biases, and external validation to confirm our assumptions



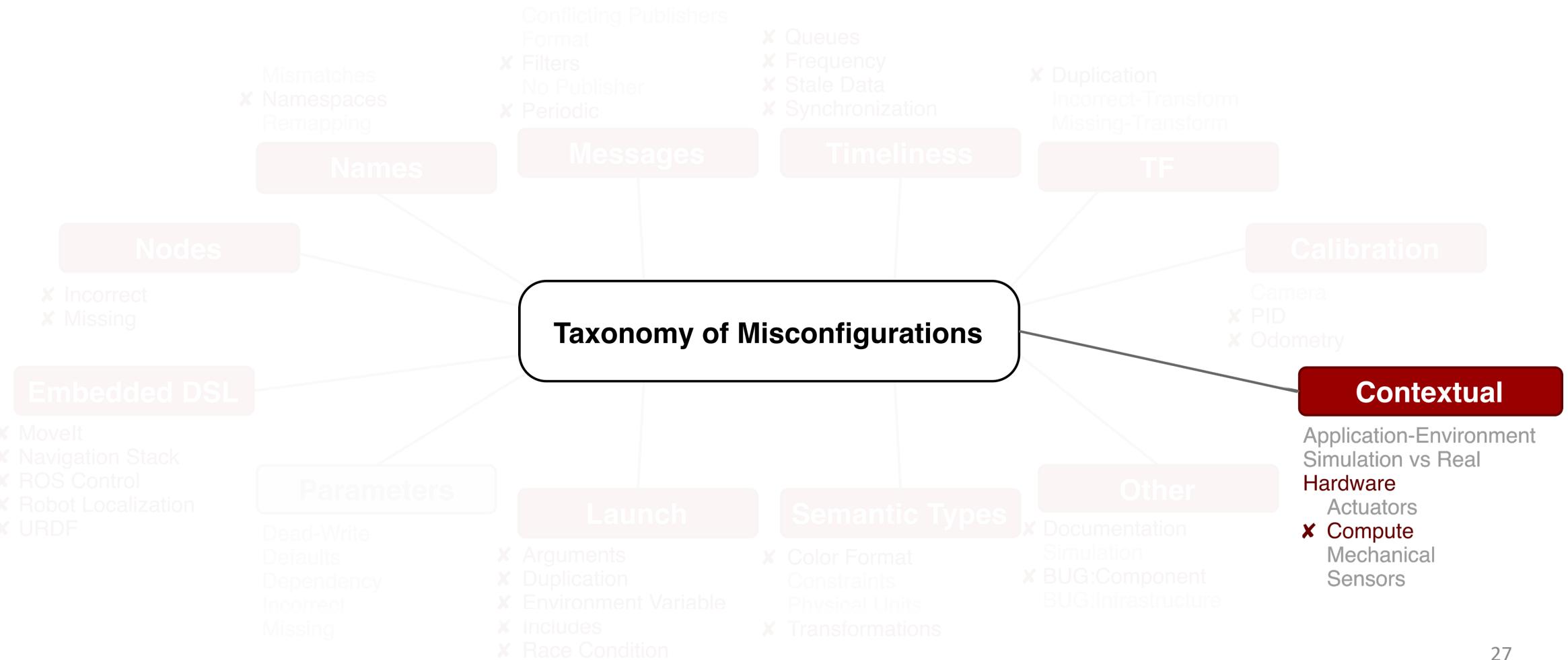
27 of 50 subcategories of misconfigurations are found yet to be addressed by current techniques



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



Contextual information is critical for analysis tools to reliably detect misconfigurations



Understanding Misconfigurations in ROS: An Empirical Study and Current Approaches

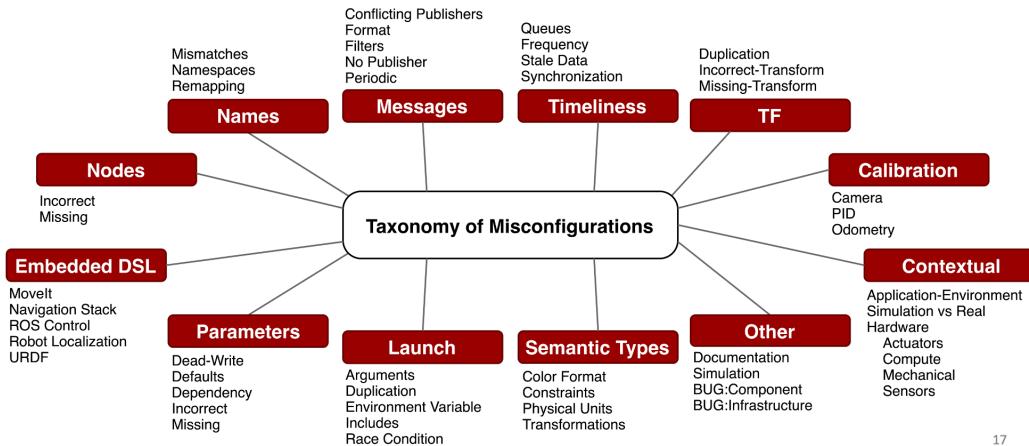
Paulo Canelas

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

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

12

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



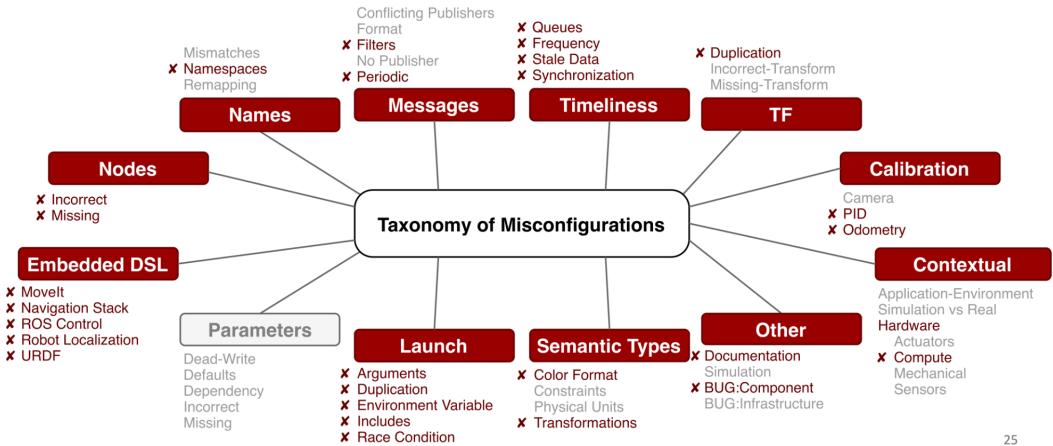
17

In this work, we study what is the broader set of misconfigurations and which tools detect 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?

27 of 50 subcategories of misconfigurations are found yet to be addressed by current techniques



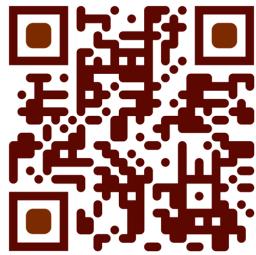
25

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.



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

In this work, we study what is the broader set of
misconfigurations and which tools detect them
Contact:
pasantos@andrew.cmu.edu

What are the gaps in current techniques address these
misconfigurations?

<https://pcanelas.com>

28 of 50 subcategories of misconfigurations are
found yet to be addressed by current techniques

