# Guidelines for Architecting Robotics Software

Ivano Malavolta[a], Grace A. Lewis[b], Bradley Schmerl[c], Patricia Lago[a,d] and David Garlan[c]

[a] *Vrije Universiteit Amsterdam, The Netherlands*
[b] *Software Engineering Institute, Carnegie Mellon University, Pennsylvania, USA*
[c] *Institute for Software Research, Carnegie Mellon University, Pennsylvania, USA*
[d] *Chalmers University of Technology, Sweden*

ARTICLE INFO

*Keywords*:
Software Architecture
Robotics
ROS

ABSTRACT

This document presents a catalog of 47 guidelines for architecting ROS-based systems. Roboticists can use our architecting guidelines for applying good design principles to develop robots that meet quality requirements.

The guidelines have been rigorously extracted from open-source ROS-based systems; the interested reader can refer to [1] for the methodological details and a detailed description of each guideline.

Contact point: Ivano Malavolta – i.malavolta@vu.nl

## References

[1] Malavolta, I., Lewis, G.A., Schmerl, B., Lago, P., Garlan, D., 2021. Mining guidelines for architecting robotics software. Journal of Systems and Software 178, 110969. URL: `http://www.ivanomalavolta.com/files/papers/JSS_ROS_2021.pdf`, doi:`https://doi.org/10.1016/j.jss.2021.110969`.

ORCID(s): 0000-0001-5773-8346 (I. Malavolta); 0000-0001-9128-9863 (G.A. Lewis); 0000-0001-7828-622X (B. Schmerl); 0000-0002-2234-0845 (P. Lago); 0000-0002-6735-8301 (D. Garlan)

# 1. Communication and Networking

In this family of concerns we group all guidelines related to the communication and networking aspects of a ROS-based system, such as types and content of messages sent/received by ROS nodes, network topology of the robotics system, frequency and rate of exchanged messages, when and how to publish/subscribe to ROS topics, etc.

**Table 1**
Guidelines for Communication and Networking (P = Provenance: GitHub | Questionnaire; $U_g$ = Usefulness score, S=Specificity: General | Specialized) | ROS-specific.

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| C1 | Use standardized ROS message formats, possibly supporting also their legacy versions. | Maintainability, Portability, Compatibility | G(9) Q(2) | 173 | R |
| C2 | ROS nodes should be agnostic of underlying communication mechanisms (*e.g.,* network protocols, deployment topology, etc.). | Maintainability, Portability | G(4) | 138 | S |
| C3 | Include health information about both nodes and data in messages containing critical data. (*e.g.,* strength of GPS signal) | Reliability | G(8) Q(1) | 94 | S |
| C4 | If the system is remotely distributed, constantly observe the status of the communication channels, hosts, and machines on the network. | Reliability | G(4) | 114 | G |
| C5 | Nodes that potentially produce/consume large amounts of messages should be configurable in terms of their publish/subscribe rates. | Maintainability, Reliability, Performance | G(3) | 107 | S |
| C6 | Selectively limit the data exchanged between nodes to provide only the information that is strictly necessary for completing tasks. | Performance, Security | G(3) | 85 | R |
| C7 | If different types of data are always sent/received together and must be synchronized, then either package them into a single message or subscribe to them using a time synchronizer filter. | Reliability | G(2) | 86 | S |
| C8 | Develop adapter components when data exchanged between nodes is not compatible (semantically), incorrect, out-of-order, or redundant. | Compatibility, Reliability | G(10) | 34 | G |
| C9 | Use services when starting up robots (instead of publishing to topics) so that the status of the system can be checked before operation. | Reliability | G(1) | 48 | R |
| C10 | Use empty messages when triggering atomic actions. | Performance | G(1) | - 34 | R |
| C11 | Frequent messages should be exchanged either via services with persistent connections or via topic-based communication. | Performance | Q(2) | - | R |
| C12 | Run multiple nodes in a single process when the overhead due to inter-process communication is too high both in terms of frequency of messages and payload. | Performance | Q(2) | - | R |
| C13 | Manage topics to avoid unnecessary publishing and subscribing. | Performance, Usability | Q(3) | - | S |

## 2. Node Responsibilities within the System

This family of concerns represents the guidelines related to the responsibilities of ROS nodes within the system, how to achieve low coupling and high cohesion between nodes, independent deployment and testing, and so on.

**Table 2**
Guidelines for Node Responsibilities within the System

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| N1 | Group nodes and interfaces into cohesive sets, each with its own responsibilities and well-defined dependencies. | Maintainability, Portability | G(3) Q(2) | 132 | G |
| N2 | Each ROS package should be responsible for one and only one feature of the system or robot capability and provide a well-defined interface. | Maintainability, Portability | G(9) | 127 | G |
| N3 | Decouple nodes with responsibilities that naturally work at different rates and use different rates for different purposes. | Performance, Reliability, Maintainability | G(7) | 123 | S |
| N4 | By design, limit unnecessary computationally-heavy operations by carefully analyzing the execution scenarios across ROS nodes. | Performance, Energy | G(6) Q(1) | 106 | G |
| N5 | Transform data only when it is used, for efficiency in terms of computation and bandwidth. | Performance, Maintainability | G(1) | 105 | G |
| N6 | Design each single node so that it is runnable (and testable) in isolation. | Maintainability | G(2) Q(2) | 88 | G |
| N7 | Provide dedicated nodes for doing introspection and querying the lower levels of the system. | Portability | G(3) Q(1) | 41 | R |
| N8 | Use a dedicated node to store and represent globally-relevant data (*e.g.,* the physical environment where the system operates) and use it as the single source of truth for all the other nodes in the system. | Reliability, Safety, Performance | G(5) | 29 | S |
| N9 | Keep the number of nodes as low as possible to support the basic execution scenarios and extend the architecture for managing corner cases. | Maintainability | Q(3) | - | S |
| N10 | Take full advantage of existing packages in the ROS ecosystem and create your own package only when it is strictly needed. | Maintainability, Reliability | Q(4) | - | S |

## 3. Internal Behavior of Nodes

The following guidelines are meant to help roboticists make architecturally-relevant decisions about internal behavioral aspects of their ROS nodes, such as spin rates, when to manage configuration errors, having stateless or stateful behavior, etc.

**Table 3**
Guidelines for Internal Behavior of Nodes

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| B1 | The behavior of each node should follow a well-defined life cycle, which should be queryable and updatable at run time. | Reliability, Maintainability, Portability | G(3) Q(1) | 70 | R |
| B2 | Nodes with high-frequency operations should be configurable so that they can operate according to available computational resources. | Performance, Reliability, Safety | G(2) | 79 | R |
| B3 | If a node is stateful and its behavior strongly depends on time and message arrival order, specify the message protocol expected by the node. | Reliability, Usability | G(2) | 84 | G |
| B4 | When possible, ROS nodes should be stateless and their behavior should not depend on previous operations or received messages. | Reliability, Usability, Maintainability, Safety | G(1) | 27 | S |
| B5 | Nodes with configuration errors should fail explicitly at bringup time. | Reliability | Q(1) | - | R |
| B6 | If a node is computationally expensive, then ensure that it only executes when it is strictly needed. | Performance, Energy | Q(2) | - | G |

# 4. Interfaces to External Users and Third-Party Developers

This family of guidelines refers to the interaction of ROS-based systems with external third-party stakeholders, mainly users and developers.

**Table 4**
Guidelines for Interfaces to External Users and Third-Party Developers Guidelines

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| I1 | Assign meaningful names to architectural elements (*e.g.*, nodes, topics, messages, services) and group them by adopting standard prefixes/suffixes. | Maintainability, Usability | G(4) Q(3) | 190 | G |
| I2 | When possible, core algorithms, libraries, and other generic software components should be ROS-agnostic. | Portability, Maintainability | G(1) Q(1) | 170 | G |
| I3 | Expose a single ROS node with interfaces for third-party users for the most common use cases. | Usability, Maintainability | G(7) | 110 | S |
| I4 | Systems interacting with other non-ROS systems should provide two types of interfaces: a ROS-independent interface for non-ROS systems and a ROS-based interface for ROS tools (*e.g.*, Rviz, Qt). | Usability, Compatibility | G(3) | 101 | S |
| I5 | Identify variation points of the system in advance, and design the system to be extended by third-parties without modifying its core nodes. | Maintainability | G(24) Q(2) | 119 | G |
| I6 | Logging should be standardized across the project and follow well-defined guidelines. | Maintainability | Q(1) | - | G |

## 5. Interaction with Hardware and Other Lower-Level Entities

This family of guidelines is about designing how ROS-nodes interact with the hardware platform (*e.g.,* sensors, actuators) in order to improve the overall quality of the system in terms of portability, maintainability, compatibility, and performance.

**Table 5**
Guidelines for Interaction with Hardware and Other Lower-Level Entities

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| H1 | Nodes interacting with simulators and hardware devices should provide identical ROS messaging interfaces to the rest of the system. | Portability | G(3) Q(1) | 197 | R |
| H2 | When possible, design the system to be hardware-independent. | Maintainability Portability Compatibility | G(15) | 165 | S |
| H3 | Decouple ROS nodes from variations in the execution environment. | Maintainability Portability Compatibility | G(9) Q(1) | 119 | S |
| H4 | State estimation nodes should support an arbitrary number and different types of sensors. | Compatibility Maintainability | G(3) | 73 | R |
| H5 | If context-specific (hardware) configuration is needed, then load it at bringup time. | Performance | G(3) | 61 | R |

## 6. Safety-Critical Concerns

In this family we report the guidelines that are strictly related to safety aspects of a ROS system, such as the management of data produced by high-frequency sensors and real-time communication.

**Table 6**
Guidelines for Safety-Critical Concerns

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| S1 | ROS nodes should be resilient with respect to the amount and frequency of data received by sensors. | Reliability | G(5) | 129 | R |
| S2 | Use different communication channels and different (hardware and software) platforms depending on the criticality and real-time requirements of the nodes. | Reliability | G(1)<br>Q(1) | 80 | G |
| S3 | For real-time requirements, collect timestamps from multiple sources (*i.e.,* do not rely on ROS-based timestamps only). | Performance | G(2) | 51 | S |
| S4 | Provide at least one globally-reachable node capable of receiving run-stop messages and stopping/resetting the whole system. | Reliability, Safety | G(3) | 49 | R |

# 7. Data Persistence

This family contains guidelines related to data persistence in a ROS system, including how to persist large raw data such as full resolution videos, how to avoid race conditions when persisting data, and others.

**Table 7**
Guidelines for Data Persistence

| ID | Guideline | Quality Requirements | P | $U_g$ | S |
|---|---|---|---|---|---|
| P1 | Avoid persisting raw data (*e.g.*, a full resolution video) if only part of it will be used. | Performance | G(1) | 97 | R |
| P2 | Avoid race conditions when persisting data received from other ROS nodes within the system. | Maintainability | G(1) | 95 | G |
| P3 | Use a dedicated node for persisting and querying long-term data. | Maintainability Performance | G(2) | 8 | S |