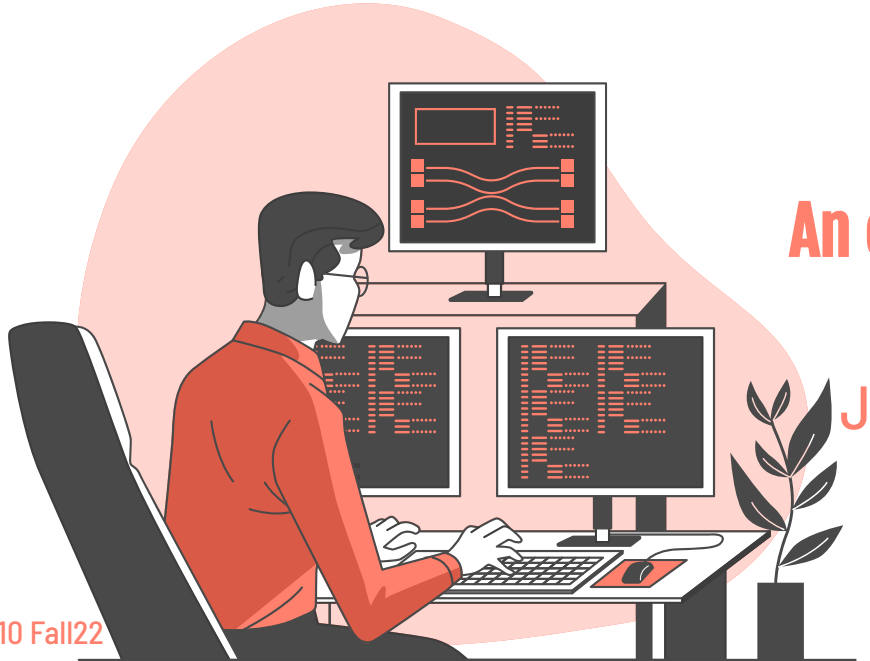
R-IDE

An environment for efficiency



CS 410 - Feasibility
Josh Peterson, Dominik Soós,
Dan Koontz, Gavin St.Clair,
James Hart, Justin Tymkin

Table of Contents

The Team

Josh Peterson
Undergraduate Researcher

Dominik Soós
Undergraduate Researcher

Dan Koontz
Undergraduate Researcher

Dr. Lee Belfore
Mentor

Gavin St. Clair
Undergraduate Researcher

Justin Tymkin
Undergraduate Researcher

James Hart
Undergraduate Researcher

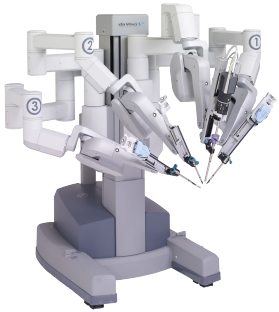
Laura Sakos
Mentor

Gavin Alberghini
Mentor

Robots & Autonomous Machines

- "Recent reports … expect that the mobile robots market is going to grow at a [rate of] 24% per year from the US$19 billion in 2018 to US$23 billion by 2021 and further more to US$54 billion in 2023”[9]

- Data published by the International Federation of Robotics suggests that the number of annual installations of industrial robots will increase from 315,000 in 2022 to 370,000 by 2024 in Asia/Australia, and from 50,000 to 60,000 in America during the same period.



Robot Operating System (ROS)

- The Robot Operating System (ROS) is an open source set of software libraries and tools that help developers build robot applications. [1]

- ROS estimated current market value
  - 2022 ≈ $270 million [2]

- ROS projected market value
  - 2028 ≈ $380 million [2]
  - 2032 ≈ $460 million [3]

Industries that utilize ROS:

- Autonomous vehicles
- Aerospace
- Health Care
- Agriculture
- Manufacturing
- Military

Some companies that use ROS-based robotics:

- NASA [4 & 5]
  - Curiosity Rover
  - VIPER program
  - Robonaut 2 [8]
- Microsoft [6]
- SONY [7]

IGVC: A collaborative effort

- ODU's autonomous vehicles utilize ROS to convert raw sensor data, perform calculations, and make intelligent decisions on the vehicles upcoming route.

- With the combined efforts of the following departments and their respective mentors, ODU hopes to place in the top 6 for the Self-drive challenge at the Intelligent Ground Vehicle Competition (IGVC).
  - Electrical and computer engineering
  - Computer science
  - Mechanical and aerospace engineering

Problem Characteristics

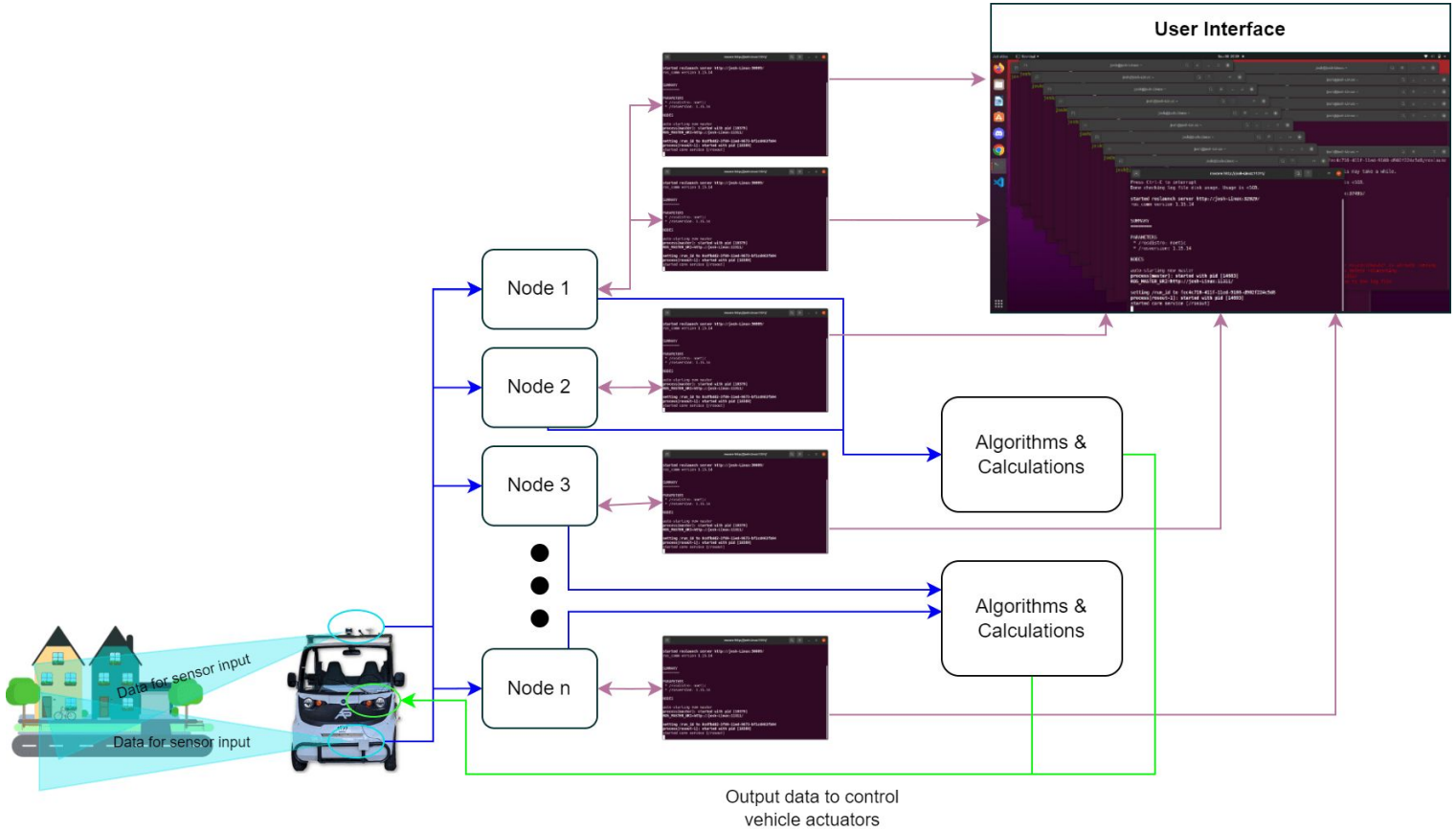
ROS workflows leverage several non-attributional windows that elevate the difficulty of debugging, monitoring & understanding

- Each ROS node has at least one terminal window associated with it
- There is no distinction when changes in one node affects a related node
- Simple debugging methodologies become time consuming due to how ROS displays errors

Many parts of the documentation are not up to date and require significant technical knowledge

Environment setup can be difficult in any project, but especially in one that can use multiple versions and distributed dependencies

Current Developer Interface Diagram



ROS development contains high barriers of entry for new developers and environments.

Our Customers

Industry

Hobbyists

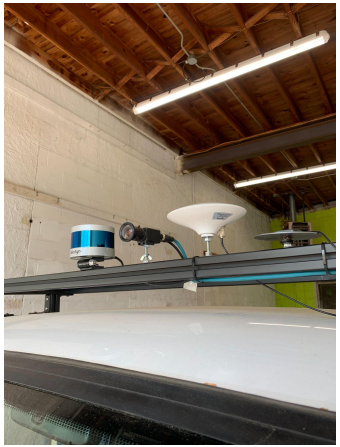
Universities

ROS Nodes

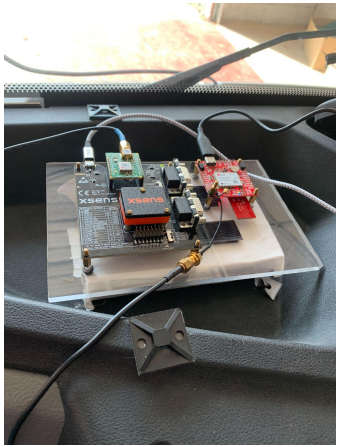
- A node is a process that performs computation, an executable file within a ROS package.
- ROS nodes are combined together in a graph and communicate with each other using topics, services, and Parameter server.
- To create a robot application with ROS requires many nodes. The LiDAR alone requires 3-5 nodes and services.

Sensors for Monarch 1

Weatherproof
OPS243C
Radar

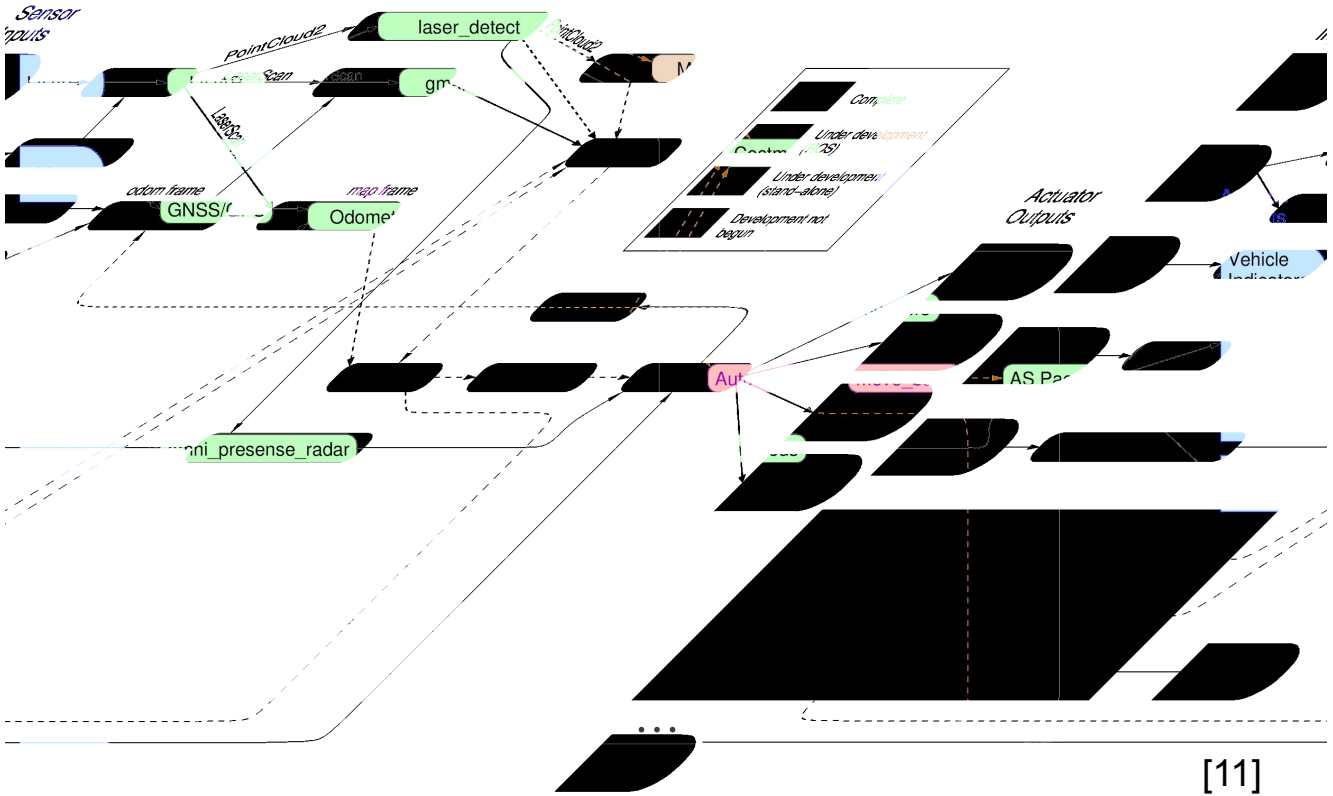


LiDAR
Wide FOV
Camera



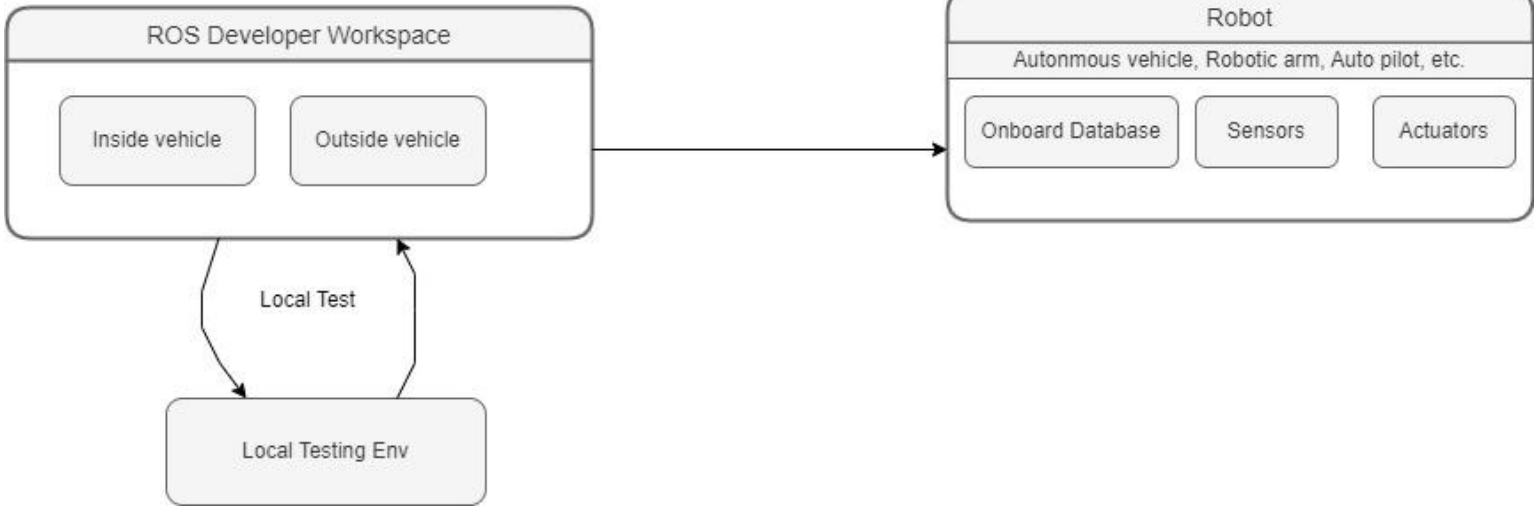
ublox
MTI-670 IMU

Architecture for Monarch 1

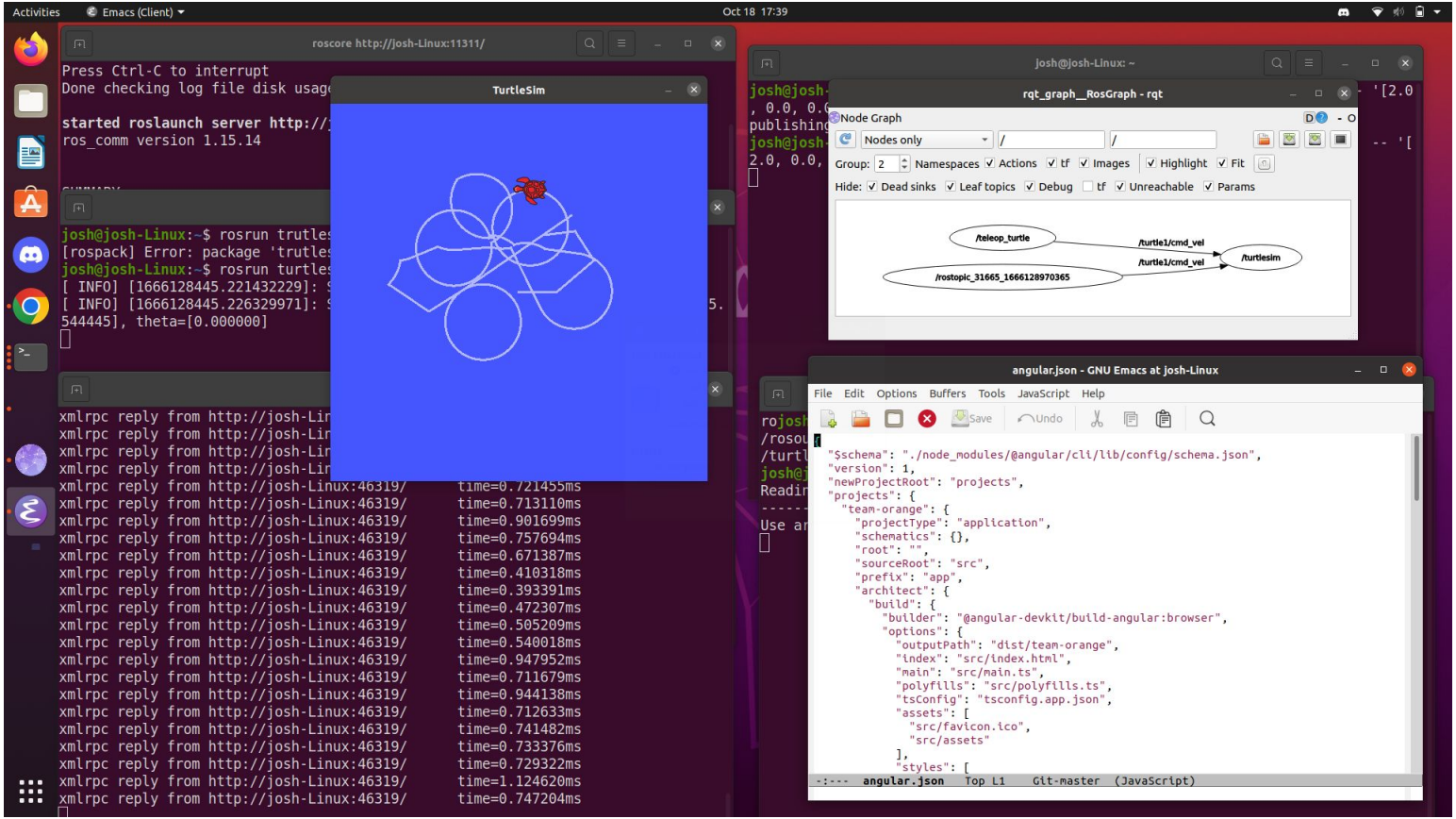


[11]

Current Process Flow



Current Developer Interface



Solution Characteristics

01

Simplify UI

Create a GUI that simplifies common tasks and commands and accesses visualization tools

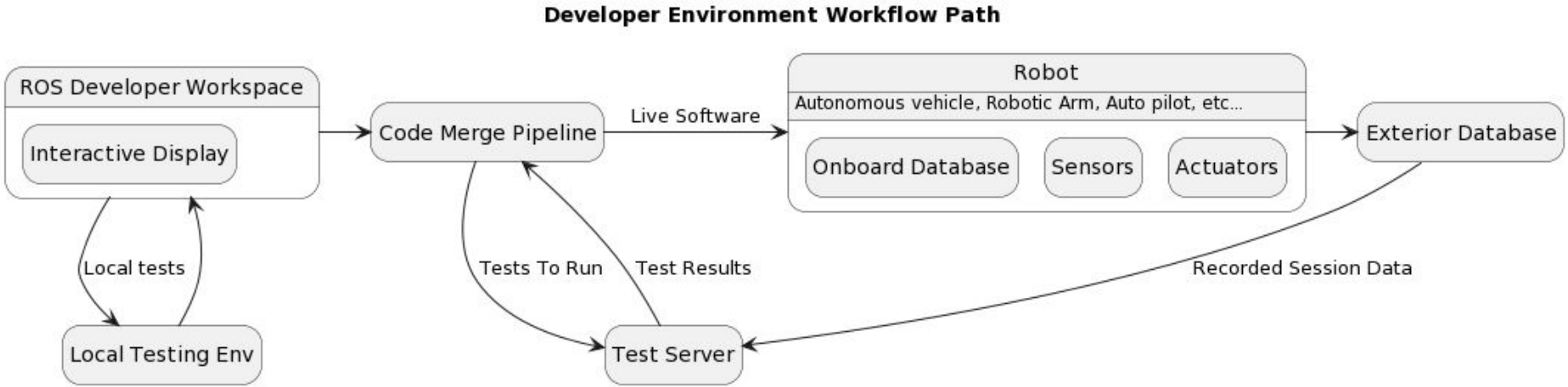
02

Quick Environment Setup

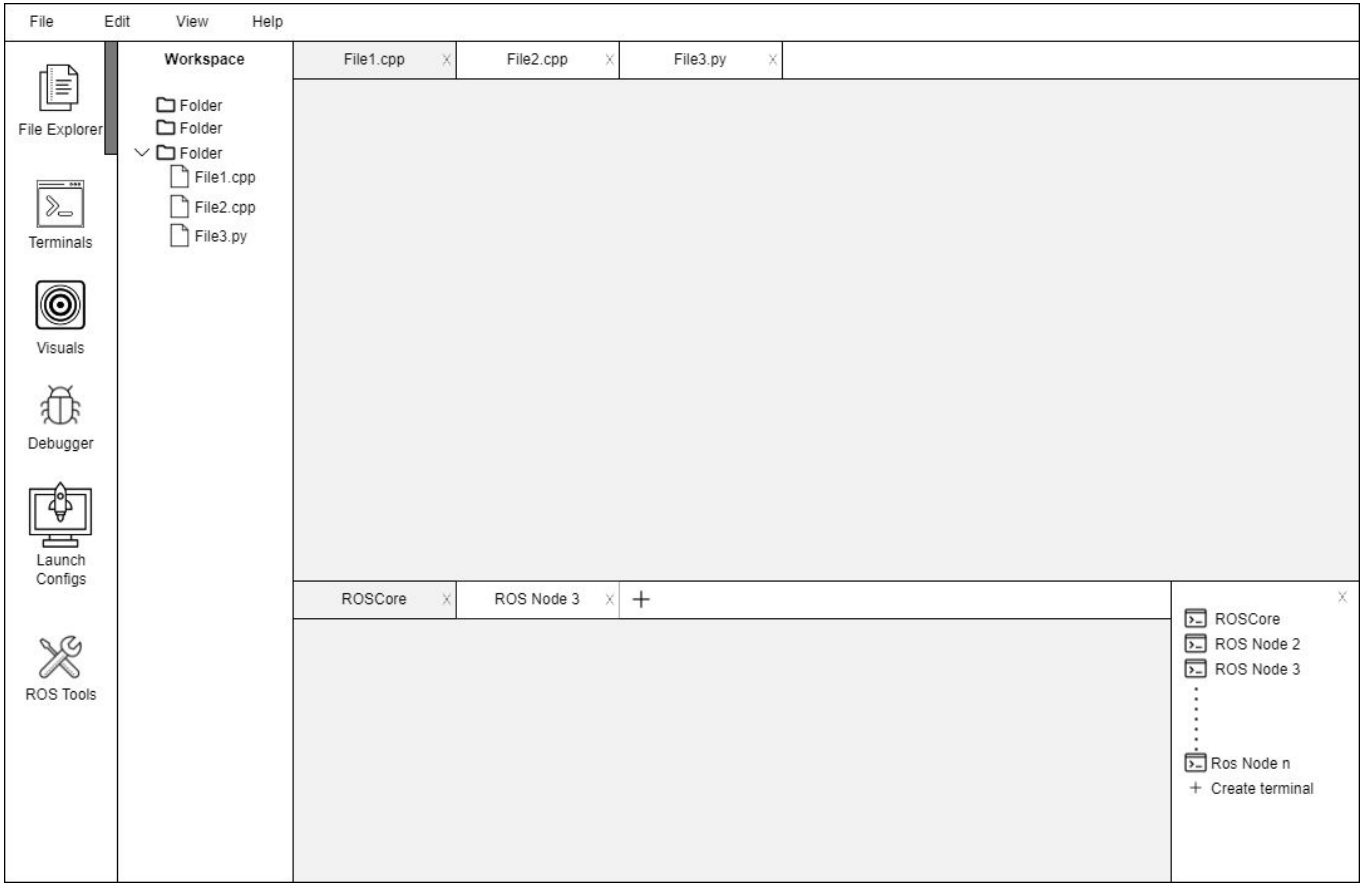
Create a process to quickly build an environment for any ROS project existing or new

An approach to simplify and speed up the development lifecycle and learning process.

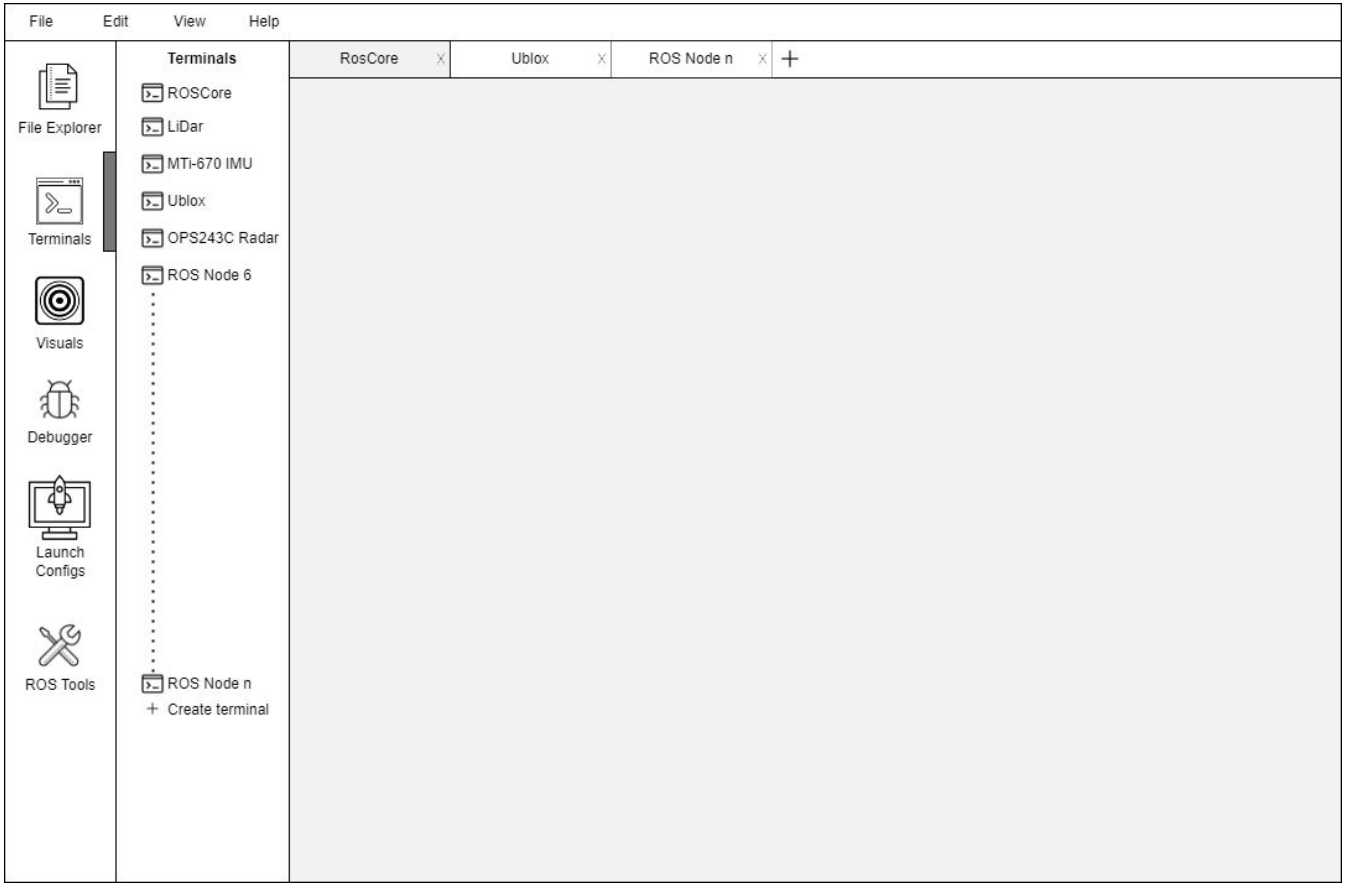
Solution Process Flow



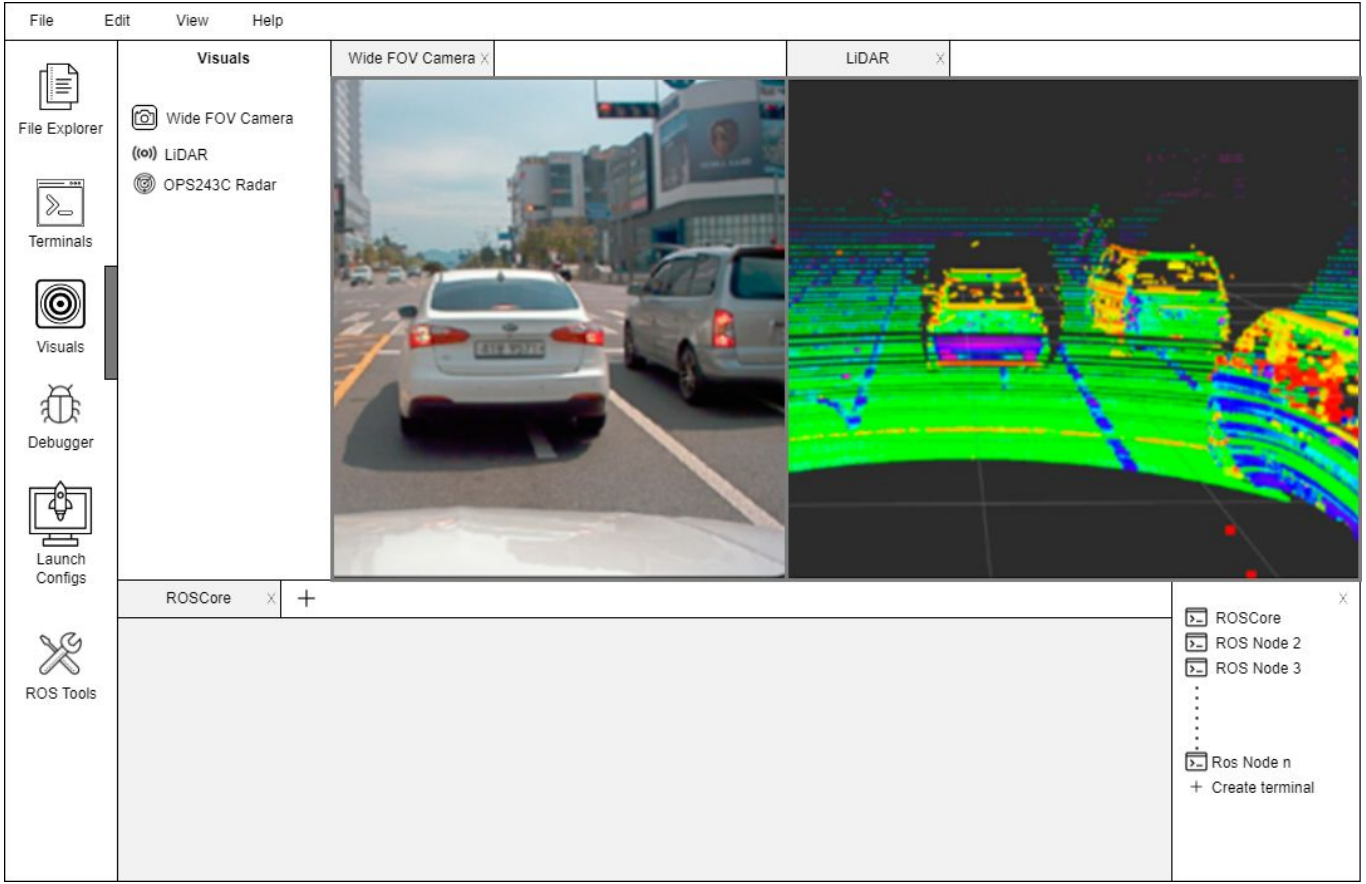
GUI: File Explorer & Text Editor



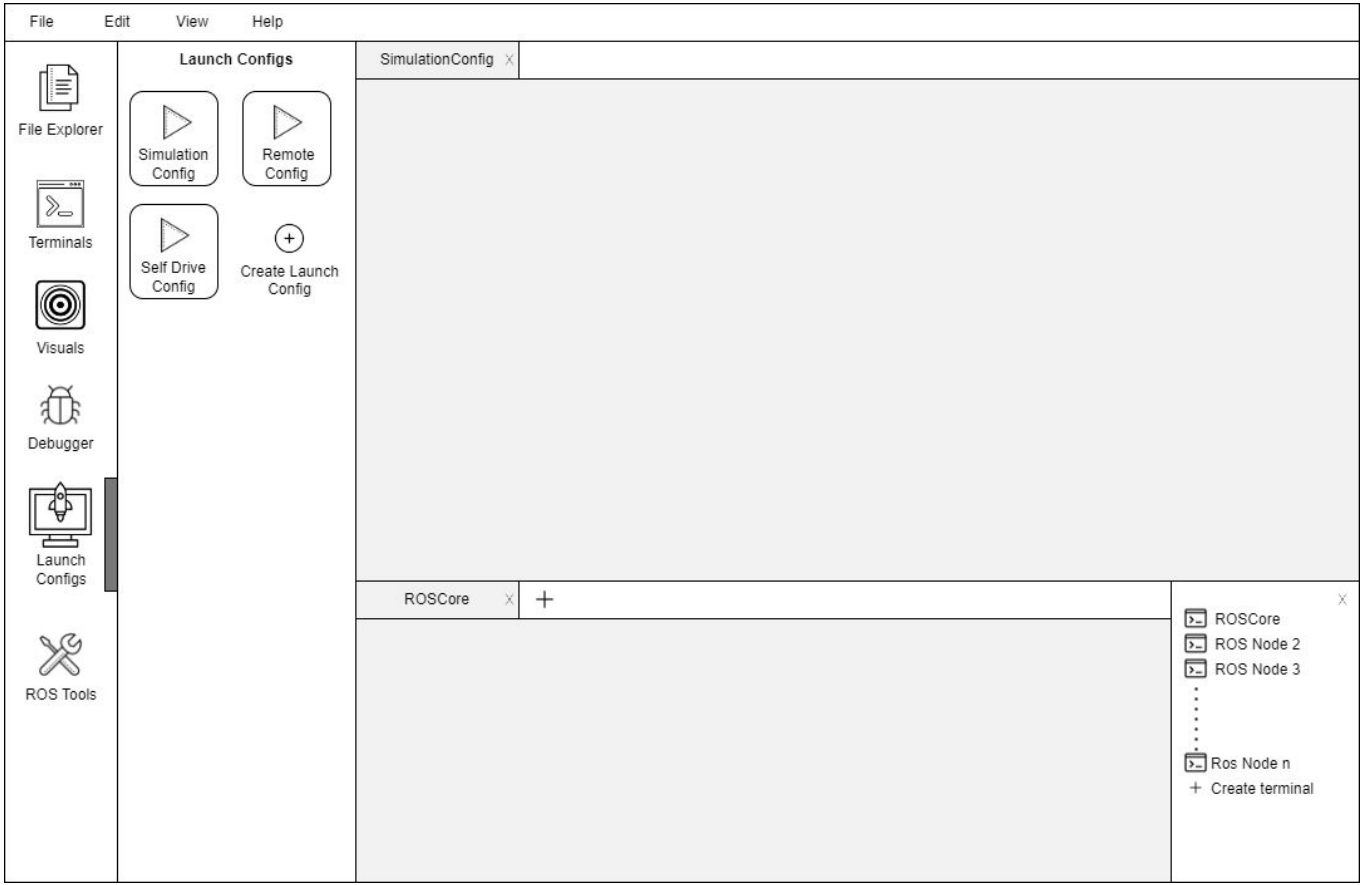
GUI: Terminal View



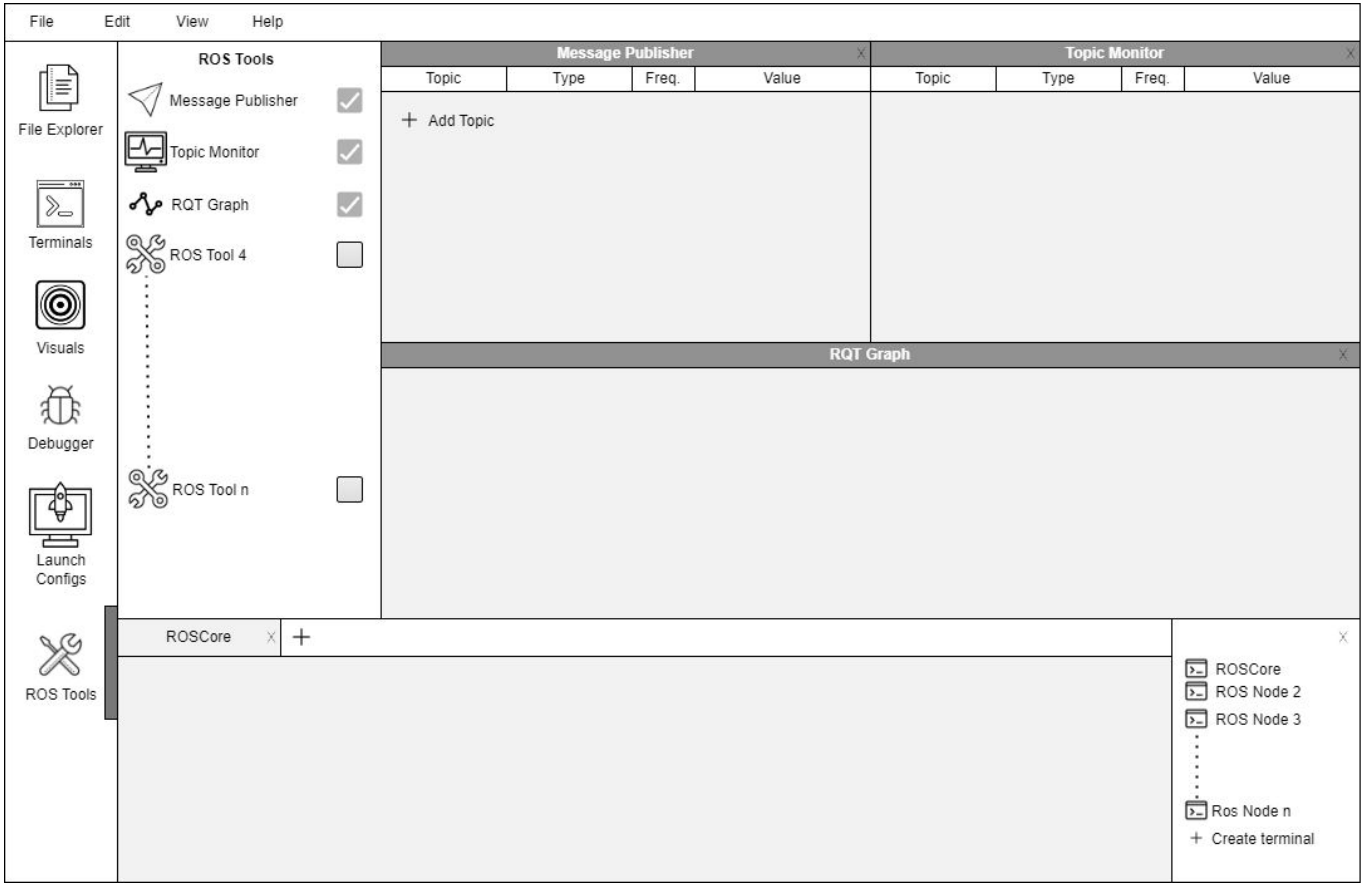
GUI: Visuals


File
Edit
View
Help
File Explorer
Terminals
Visuals
Debugger
Launch Configs
ROS Tools
Visuals
Wide FOV Camera
LiDAR
OPS243C Radar
Wide FOV Camera
LiDAR
ROSCore
ROSCore
ROS Node 2
ROS Node 3
Ros Node n
Create terminal

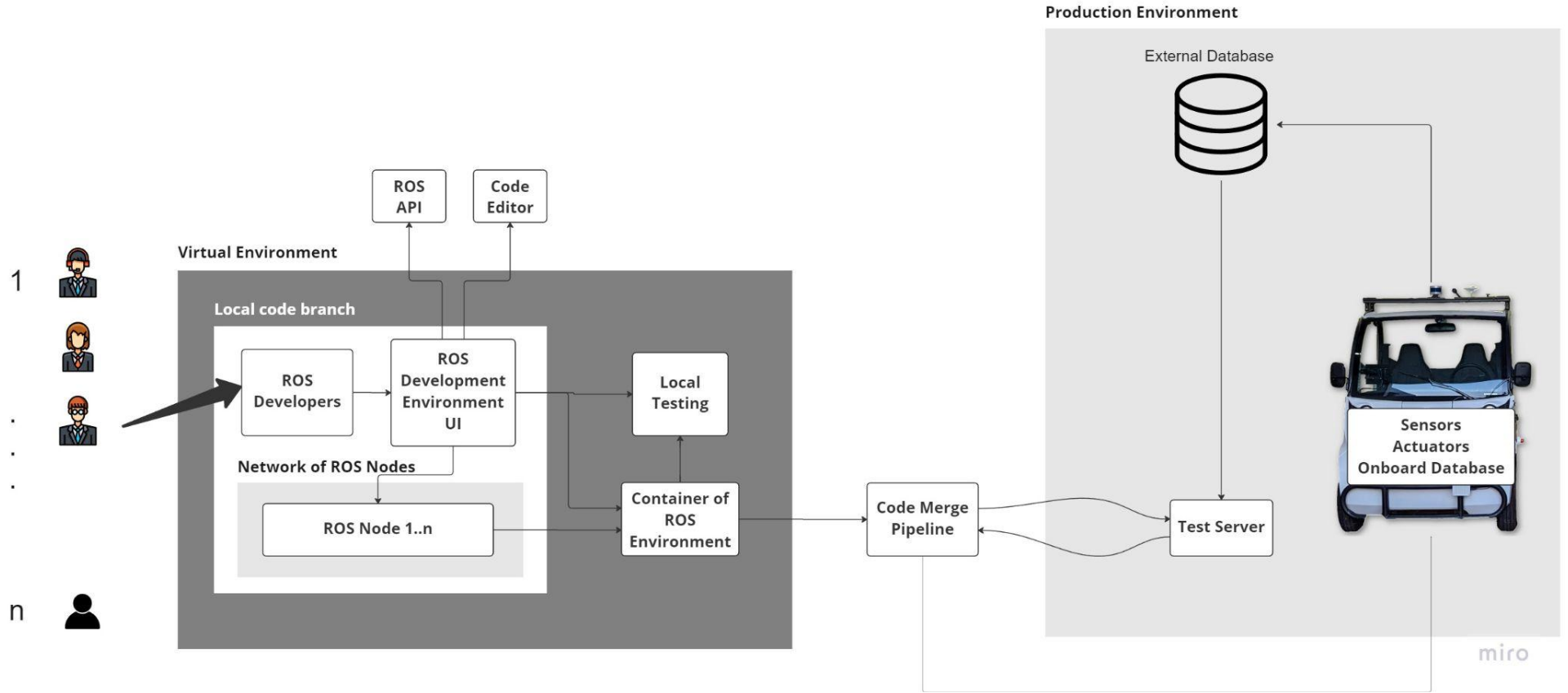
GUI: Launch Configs



GUI: ROS Tools



Major Functional Component Diagram



GUI Competition Matrix

	R-IDE	RQT
Dockable Windows	Yes	Yes
Modular Plugins	Yes	Yes
ROS2 Support	Yes	Yes
Debugger Support	Yes	No
RViz Support	Yes	Yes
OS Support	Yes	Yes
ROSTopic support	Yes	Yes
ROS bag support	Yes	Yes
Web page View	Yes	No
Launch nodes from GUI	Yes	From Launch File
Perspective Shifting	Yes	Yes
Most Recent Update (< 6 months)	In Active Development	2019?

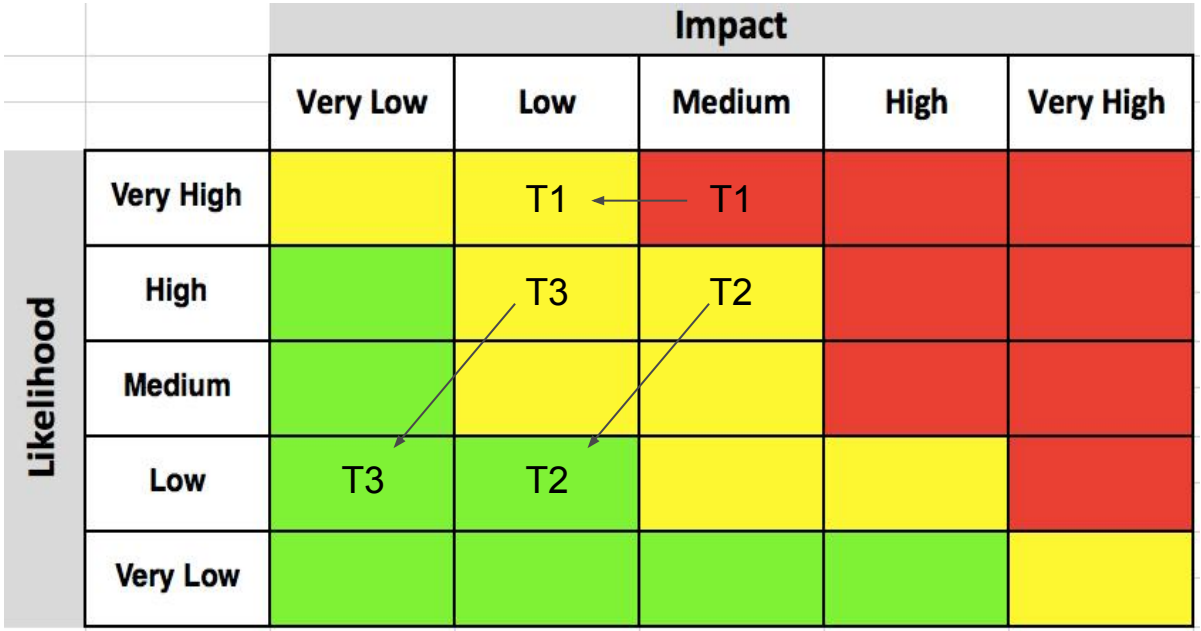
IDE Indirect Competition Matrix

IDE Competition Matrix

	R-IDE	CLion	VSCode	Emacs	VIM	QT Creator	Eclipse
Debugging	Yes	Yes	Yes	Yes	No	Yes	Yes
Parallel Terminals	Yes	Yes	Yes	No	No	Yes	Yes
Manually Rename Terminal	Yes	Yes	Yes	No	No	???	No
Auto-Naming Terminals	Yes	No	Not Accurately	No	No	No	No
File Exploring GUI	Yes	Yes	Yes	Yes	No	Yes	Yes
Autosave Feature	Yes	Yes	Yes	Yes	No	Yes	Yes
Free to use	Yes	Free Educational Licenses	Yes	Yes	Yes	No* *Free under the (L)GPL and expected to contribute to the Qt project.	Yes
Most Recent Update (< 6 months)	In Active Development	10 Oct, 2022	September 2022	18 February, 2022	23 August, 2021	23 March, 2022	29 July, 2022

miro

Technical Risk Matrix



T1: Low computational power

- **Risk**: Computationally expensive tasks such as LIDAR
- **Mitigation**: Offload to server or faster computer, Ubuntu as main OS

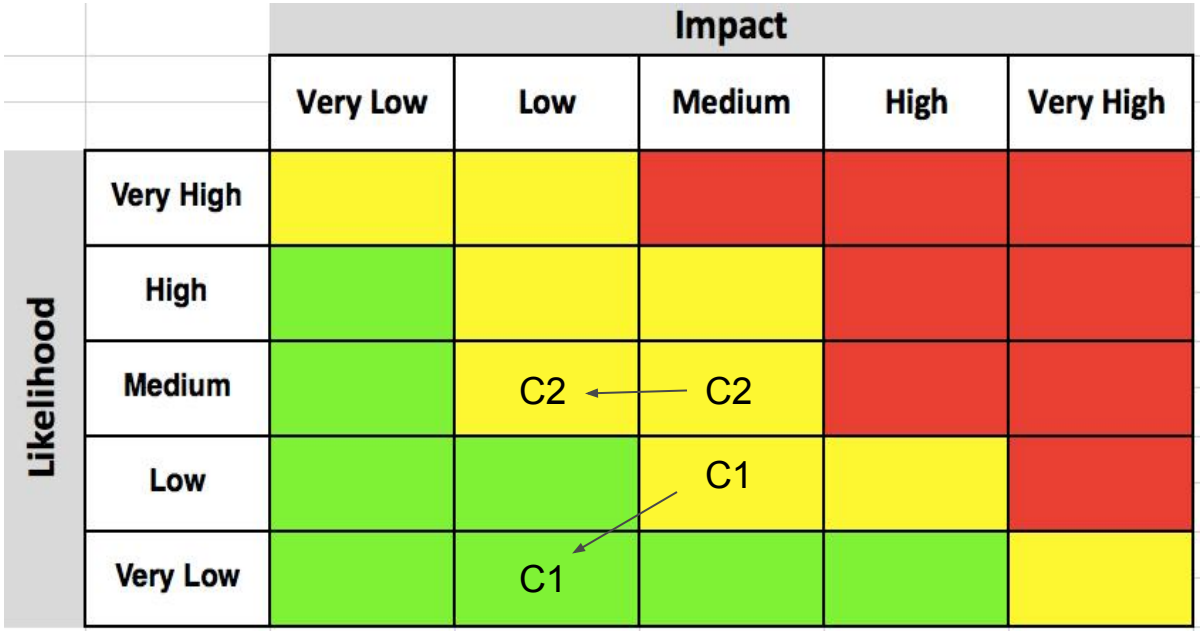
T2: Error detection

- **Risk**: The source of error messages are unknown
- **Mitigation**: Debugging feature

T3: ROS1 → ROS2

- **Risk**: Transitioning project from ROS1 to ROS2
- **Mitigation**: Using ros1_bridge, which allows bidirectional communication

Customer Risk Matrix



C1: Shifting problem statement

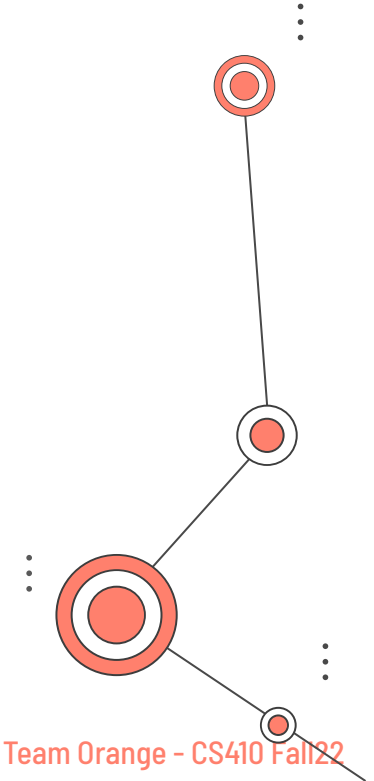
- **Risk**: constant changing problem statement and characteristics
- **Mitigation**: coming to a better understanding of the problem

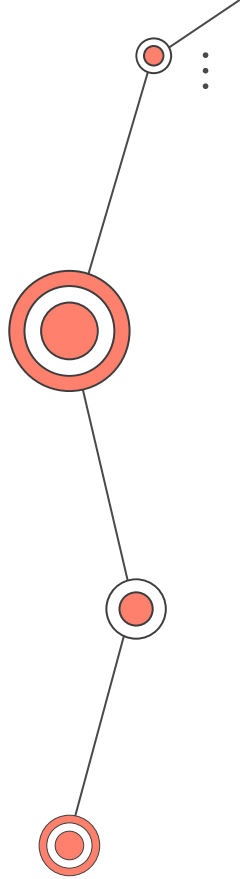
C2:

- **Risk**: customers do not understand how to use product
- **Mitigation**: documentation and well-written user requirements

Thanks!

Do you have any questions?





References

Citation

1. "Robot Operating System." ROS, www.ros.org/.
2. "Global Robot Operating System Market Research Report 2022(Status and Outlook)." Market Reports, www.marketreportsworld.com/global-robot-operating-system-market-21185690.
3. "Robot Operating System Market." Future Market Insights, www.futuremarketinsights.com/reports/robot-operating-system-market.
4. "Lunar Rover." Open Robotics, www.openrobotics.org/customer-stories/lunar-rover.
5. Wessling, Brianna. "Open Robotics Developing Space ROS with Blue Origin, NASA." The Robot Report, 11 Feb. 2022, www.therobotreport.com/open-robotics-developing-space-ros/.
6. Ackerman, Evan. "Microsoft Announces Experimental Release of ROS for Windows 10." IEEE Spectrum, IEEE Spectrum, 24 June 2021, spectrum.ieee.org/microsoft-announces-experimental-release-of-ros-for-windows-10.
7. Tomoya Fujita. "aibo with ROS", https://roscon.ros.org/2018/presentations/ROSCon2018_Aibo.pdf
8. Brian Gerkey on September 1, 2014 8:07 AM. "ROS Running on ISS." Ros.org, www.ros.org/news/2014/09/ros-running-on-iss.html.
9. M. Cardona, A. Palma and J. Manzanares, "COVID-19 Pandemic Impact on Mobile Robotics Market," 2020 IEEE ANDESCON, 2020, pp. 1-4, doi: 10.1109/ANDESCON50619.2020.9272052. https://ieeexplore.ieee.org/abstract/document/9272052
10. International Federation of Robotics. Annual installations of industrial robots 2019-2020 and 2021*-2024* [Graph]. Retrieved from https://ifr.org/downloads/press2018/2021_10_28_WR_PK_Presentation_long_version.pdf
11. "Wiki." Ros.org, wiki.ros.org/Security.
12. Dieber, Bernhard, et al. "Security for the robot operating system." *Robotics and Autonomous Systems* 98 (2017): 192-203.

References Cont.

Competition Matrix References

1. CLion IDE: https://www.jetbrains.com/clion/
2. CLion Hatchery: https://plugins.jetbrains.com/plugin/10290-hatchery
3. CLion ROS Support: https://plugins.jetbrains.com/plugin/11235-ros-support
4. VS Code: https://code.visualstudio.com/
5. QT Creator with ROS: https://github.com/ros-industrial/ros_qtc_plugin/blob/devel/gh_pages/index.rst
6. Emacs with Rosemacs: http://wiki.ros.org/rosemacs
7. VIM ROS: https://github.com/taketwo/vim-ros
8. TMUX: https://github.com/tmux/tmux
9. GNU Screen: https://www.gnu.org/software/screen/
10. Terminator: https://github.com/gnome-terminator/terminator
11. "Software Design Report for the ODU Monarch I", Version 1.3, Jul 8, 2022, page 3. Author Lee Belfore"

Glossary

https://www.cs.odu.edu/~410orang/#/glossary

ROS - The Robot Operating System (ROS) is an open source set of software libraries and tools that help developers build robot applications.

Autonomous Machine - A machine capable of sensing its environment, carrying out computations to make decisions, and performing actions in the real world.

ROS Bag - A bag is a file format in ROS for storing ROS message data. Bags are the primary mechanism in ROS for data logging, which means that they have a variety of offline uses.

ROS Topic - Topics are named buses over which nodes exchange messages. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption.