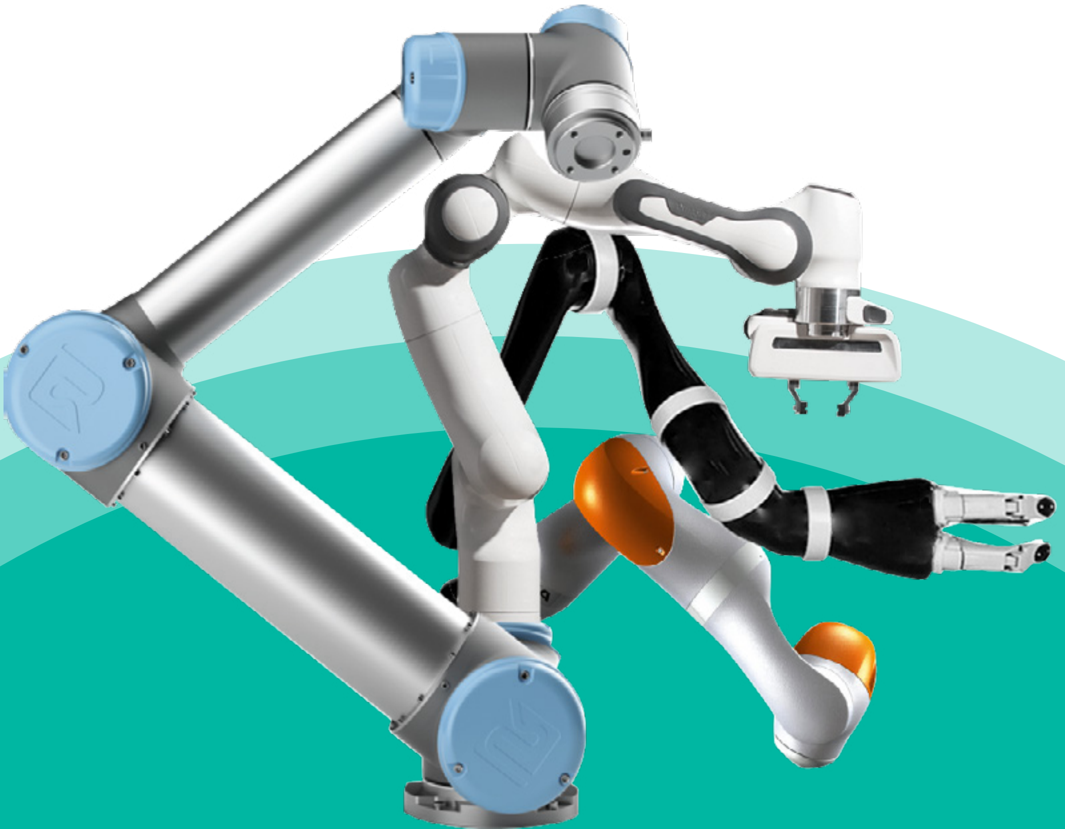
How to leverage **Open Source Robotics** & protect your company’s IP



www.picknik.ai

Many companies are entering the exploding robotics market and are looking to leverage open source technology to quickly deliver value to their stakeholders. Approximately 90% of organizations report using open source software today[1], and 99% of IT professionals indicate that open source software is somewhat to extremely important in the enterprise[2].

Just as the web has been an incredible opportunity for entrepreneurs to quickly build amazing companies off open source internet technologies, open source is opening doors for the next generation of robotic startups and research groups. Yet we at **PickNik Robotics** often see companies struggling with how to best use these open technologies and where to draw the line between open source and their core intellectual property.

– Dave Coleman

"Approximately 90% of organizations use open source software today[1]"

Open Source Robotics Software

Key Advantages

There are many advantages of leveraging popular open source projects like the **Robot Operating System (ROS)**; a few of them are reduced cost, easier hiring, tested industrial standards, and avoiding vendor lock-in. Rather than reinventing the wheel for each company's robotics platform from the ground up, you can leverage vetted open source software to save millions in R&D spending. Popular open source frameworks typically have had hundreds of developers review and perfect the code. **Open source software avoids vendor lock-in**, as any capable development team can modify the software as needed.

Highly-skilled roboticists are in strong demand, with the required breadth of mechatronics, artificial intelligence, and software engineering commanding high salaries and long onboarding times. If you use popular open source projects, you can more easily tap into fresh talent from university robotics programs around the world who already know your tech stack.

Key Disadvantages

Despite numerous advantages, there is no free lunch. Just like any typical closed-source software package, there are high quality and low quality open source projects. Projects like ROS are often loosely federated, meaning anyone can contribute components with mixed levels of support. A common risk of open source robotics is inheriting technical debt from a project that your team is unable to fix due to lack of expertise or familiarity. **No software is initially turn-key for your company's** unique applications and requirements, although with time, it can be customized in various ways.

[1] "As Open-Source Adoption Skyrockets in Enterprise, Linux Addresses Ease of Use," Marlene Den Bleyker, SiliconANGLE, 2 June 2017.

[2] "The State of Enterprise Open Source," a Red Hat Report by Illuminas, April 2019.

How To Best Leverage Open Source

From our experience deploying open source robotics across many industries and applications, here are our top recommended strategies to best leverage open source:

1

Ensure the project uses a permissive license.

Check that the open source software license is non-copyleft: it guarantees the freedoms to use, modify, and redistribute, but also permits proprietary derivative works. Common ones include BSD, Apache2, and MIT licenses.

2

Avoid creating your own version of the project's code ("forking").

This makes it more difficult to keep up with future improvements within the original project. Software's natural tendency is to 'rot' due to the everchanging upgrades to computer platforms. Forking will create a large maintenance burden for your team and isolate your fork from future project upgrades.

3

Only use modular projects that allow you to easily add extensions.

Your team will eventually need to switch out components for custom solutions to your application. Ensure the software you leverage has plugins, hooks, and other distributed approaches in its design.

4

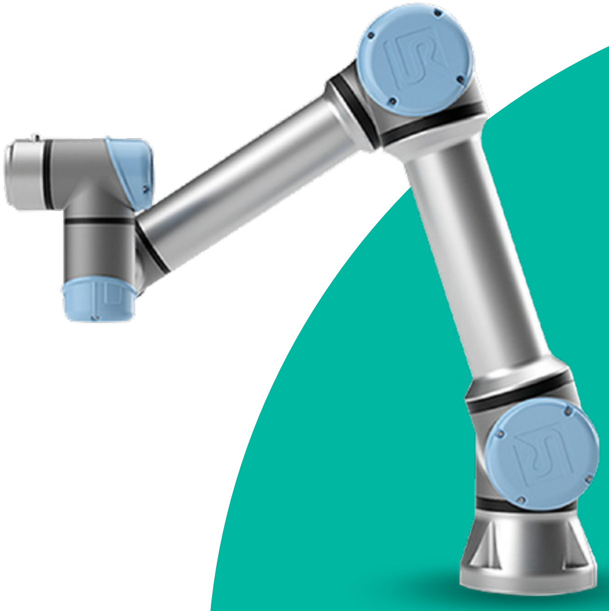
Be wary of projects not backed by a company or team.

Orphaned and abandoned projects have no one to support bugs and may be a dead end. A company providing business-class support signals that someone is at the helm keeping an eye on architecture and quality.

5

Work with the active developers or original authors of the project who know the code best.

These passionate contributors to open source can solve your problems much faster (and in their sleep!).



How PickNik Contributes to Open Source While Protecting Our Clients’ IP

PickNik is an advocate for open source robotics and contributes a portion of our projects’ work back to open source, while still protecting our clients’ core IP. Our primary focus is serving our clients in strategically utilizing cutting-edge open source robotics software. We de-risk open source software usage by providing them expert software development and support.

The team at PickNik is actively developing a number of popular open source packages including the MoveIt Motion Planning Framework, a key application in the ROS ecosystem. We consider MoveIt our core product and part of our background technology, and it also happens to be royalty free and open sourced under a permissive license (BSD). We maintain a product roadmap for MoveIt that we share publicly to ensure alignment of all contributors.

This is the best approach we have found to contribute to open source while protecting our client’s IP:

All custom software development not using open source is 100% our client’s IP.

We are happy to do closed source work if their tech stack does not involve extending open source projects.

All robot-specific and application-specific logic is 100% our client’s IP.

This software goes into clearly delineated, client-specific software packages and is closed-source.

We open source bug fixes, maintenance, and non-core tech features

This applies to existing open source projects only. An example of a small feature is speeding up an existing section of code or making a parameter user-configurable.

We avoid forking existing open source projects.

There are many creative ways to keep our client’s core-technology and IP separate without creating your own version of a project’s code. For example, MoveIt has an extensible plugin-architecture that enables us to add custom proprietary code without having to change the installed version of MoveIt. This ensures that our client’s project benefits from future bug fixes and development efforts.

Why We Avoid Forking Existing Open Source Projects

A fork of an existing project is when a company decides to not track the public, main version of an open source project but instead decides to break away and develop their own version. The typical motivation for this is a company's desire to ensure no intellectual property is shared with the outside world, thereby potentially gaining a competitive edge. Somewhat unintuitively, we do not believe this is in a company's best interest for a number of reasons:

- **A company's fork will miss out on all new features & optimizations being added to the parent project.**
- **A company's fork will require the company to do full maintenance of the software thereafter.**
- **It is difficult to get help from the current and original authors of the open source project to help fix bugs on a fork of their own project.**

Takeaways

A company can gain tremendous value from open source projects and the benefits far outweigh the limitations. We have presented common best practices for how to leverage open source.

At PickNik, we are confident that open source can be leveraged while protecting our client's IP. Contributing back to open source makes sense from a practical standpoint, given what one loses if they fork a project.

Beyond that, contributing back fosters good will in the community, encourages others to contribute, and is just the right thing to do. Through actions like these, we believe everyone can gain from open access to the economic prosperity promised by robotics.