As low-cost hardware becomes ubiquitous, so increases the potential for large-scale robotic impact. LoCoBot is the newest addition to the affordable-yet-performant family of robots, which has been the driving factor behind Duckietown since the first MIT class ran in 2016. The reach of Duckietown, and now the AI Driving Olympics (AI-DO), spans engineering, research, and education, bringing the world an affordable way to learn about autonomous driving. PyRobot and LoCoBot stand to serve the same role for the space of manipulation and home service robotics, and our proposal highlights a unique opportunity for both organizations. In this proposal, we describe how **unifying the PyRobot and AI-DO ecosystems** can be mutually beneficial to both PyRobot and Duckietown in terms of: new hardware possibilities and educational opportunities, as well as new benchmarks, datasets, and baselines.

Just like PyRobot, we've built our software with robotics in mind. However, PyRobot at present still requires custom installation, which is prone to user-specific failures and mistakes. With the help of AWS and Aptiv, all of Duckietown's software has been containerized, allowing competitors and students to focus less on the frustrations of robotics (installation, networking, hardware interfacing, etc) and more on the software implementations. To move towards a unified API, we'd work with FAIR to containerize PyRobot and bridge the two codebases, enabling both sets of robots to run with the same interface.

For the AI-DO, competitors are given RL, Imitation Learning, and classical robotics baselines (across Pytorch, Tensorflow, and ROS), allowing them to better understand how the API works. By standardizing the interfaces, not only does PyRobot gain access to all of the AI-DO baselines, but also hundreds of terabytes of log files that can be used to train vision and control modules. These same codebases are used to drive our educational efforts (now running globally in five universities). Standardization allows us to provide a "classroom-in-a-box" to any educator, complete with lectures and corresponding Python notebooks for practical implementation. These can easily be fashioned into lectures and exercises about robotic manipulation, a subject on which many fewer open-source educational resources exist.

With containers, we can run the same algorithms across different simulators and hardware, all without significant user debt. Most excitingly, this unification paves the way towards **large-scale, standardized robotic evaluation.** In machine learning, the lack of standardized benchmarks makes it difficult to gauge progress, especially when hardware interfaces are limited to industry and a few academic institutions. At Duckietown, we've built the world's first open-source *Robotarium,* a fully-containerized, automated, hardware testing facility that allows us to test user and student submissions to the AI-DO. By maintaining the same API, we can help develop the same infrastructure for PyRobot's supported robots, culminating in a hardware-based benchmark that can become the defacto standard for evaluating planning, learning, and control algorithms.

In summary, if our proposal is selected, we will work to unify the infrastructure of Duckietown (and the Duckiebots) and PyRobot (and the LoCoBots). We believe that both Duckietown and the PyRobot project stand to benefit from our proposed unified approach.