



# Movelt Roadmap

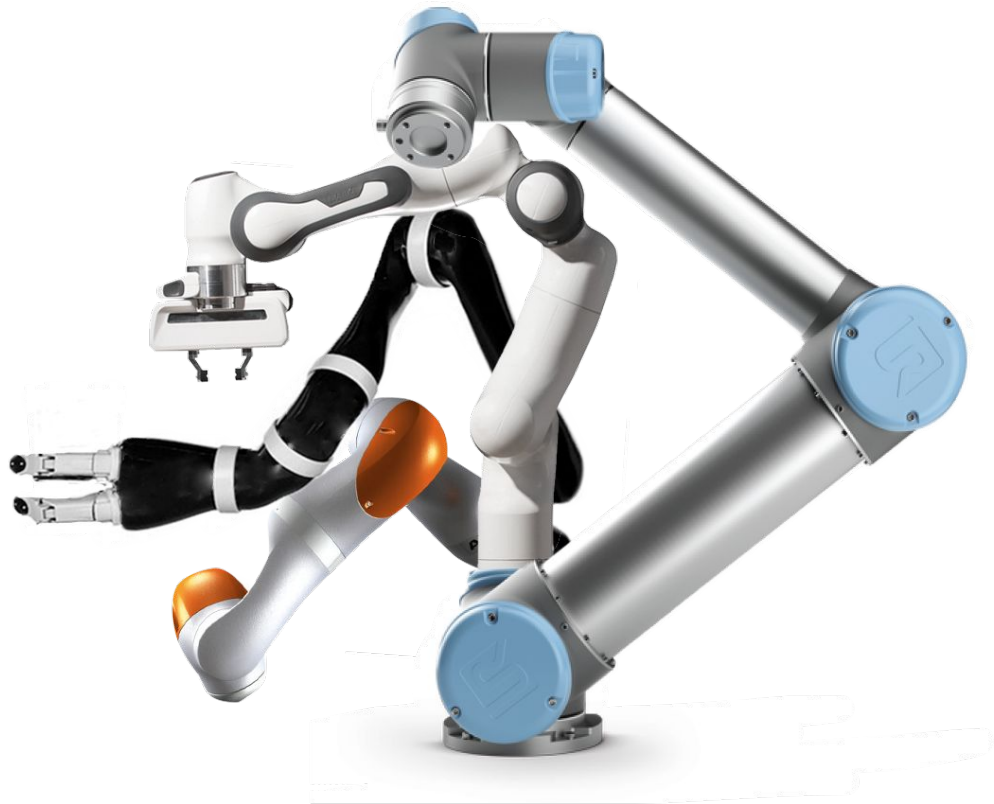
Movelt 2.0 and Upcoming Features



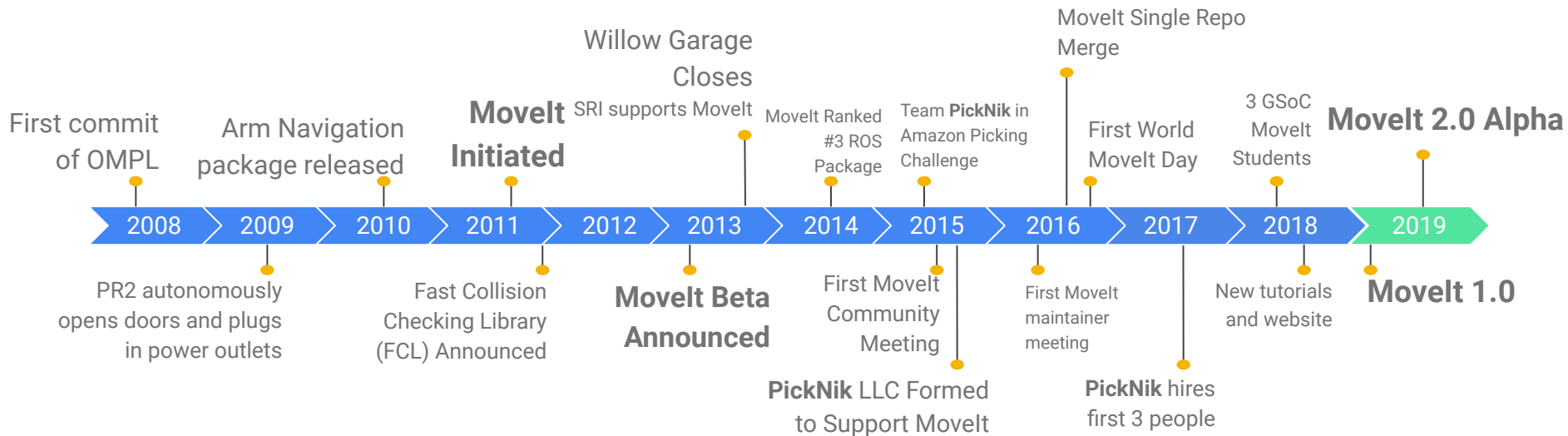
PICKNIK

# Outline

- Timeline
- Why ROS 2
- MoveIt 2.0 Alpha by Acutronics
- Current state of MoveIt 2.0 Alpha
- Future Roadmap and Development
- MoveItCpp



# Timeline



---

# Why ROS 2

Features and Vision



## Why ROS 2?

- Deterministic and real-time compliant
- Multi-platform support
- No vendor lock-in through layered architecture
- Production-ready framework
- Open communication standards like DDS, RTPS



## Fully Leverage ROS 2

- Lifecycle management of MoveIt nodes
  - Deterministic startup, reset & shutdown sequences
- Leverage ROS 2 component nodes
  - Ability to run MoveIt as single or multi-process
  - Replace pluginlib with components
- Cleanup API
  - More generic and standalone interfaces



# Improved Interfaces / State Machines

- Deprecate the Pick and Place pipeline
  - Fully support the MoveIt Task Constructor
- First class support of state machines
- Non-ROS C++ API
  - Similar to MoveGroup but without middleware

---

# Movelt 2.0 Alpha

Initial implementation by Acutronic





# Movel 2.0 Alpha

Announced: June 2019

ROS2 distro: Dashing Diademata

OS: Ubuntu 18.04 and Mac OS X 10.14





## MoveIt 2.0 Alpha - Progress

- Most of moveit\_core moveit\_ros ported to ROS2
- 11 external dependencies have been ported
- Functional CI infrastructure: moveit\_ci
- Capability for simple planning to joint-state goal
- Example ROS 2 control framework for Acutronic's MARA robot



---

# Movelt 2.0 Future Development

Roadmap and Milestones



# Roadmap

1. Finish migration of MoveIt 1 packages
2. Create ROS1 bridges for legacy support
3. Merge and simplify ecosystem repositories
  - a. More efficient maintenance and development
4. **Realtime Support**
5. **Deterministic Planning**



## Realtime Support

- Reactive, closed-loop control to sensor input
  - Visual servoing, octomap updates
  - Preempt motion if new collision detected
- Separate global and local planner (hybrid planning)
  - Global planner (full collision checking): ~30Hz
  - Local Planner (IK-based, field-based): ~300Hz
- Zero-memory copy integration to controllers (ros\_control)
  - Tighter integration to ros\_control
- Integrate pilz\_industrial\_motion



# Deterministic Planning

- Out of box / default planners return reliable paths
  - Improved support for OMPL, TrajOpt
- Further optimize or smooth motions
  - Default use TOTG, TOPP time parameterization
  - Post-processing optimization (STOMP, TrajOpt)
- Fully featured Cartesian Planner



# Future development

Goal: Beta version by Q1 2020

Project Funding: ROSin FTP + PickNik





---

# MovelCpp

“Non-ROS” low-level MoveIt API



# MoveItCpp

- As simple as MoveGroup
- Disabling performance-bottleneck ROS
- Direct access to (custom) core components
- Multi-robot support
- Industry-directed applications
- Faster product development

PR: <https://github.com/ros-planning/moveit/pull/1656>

---

# Open Discussion