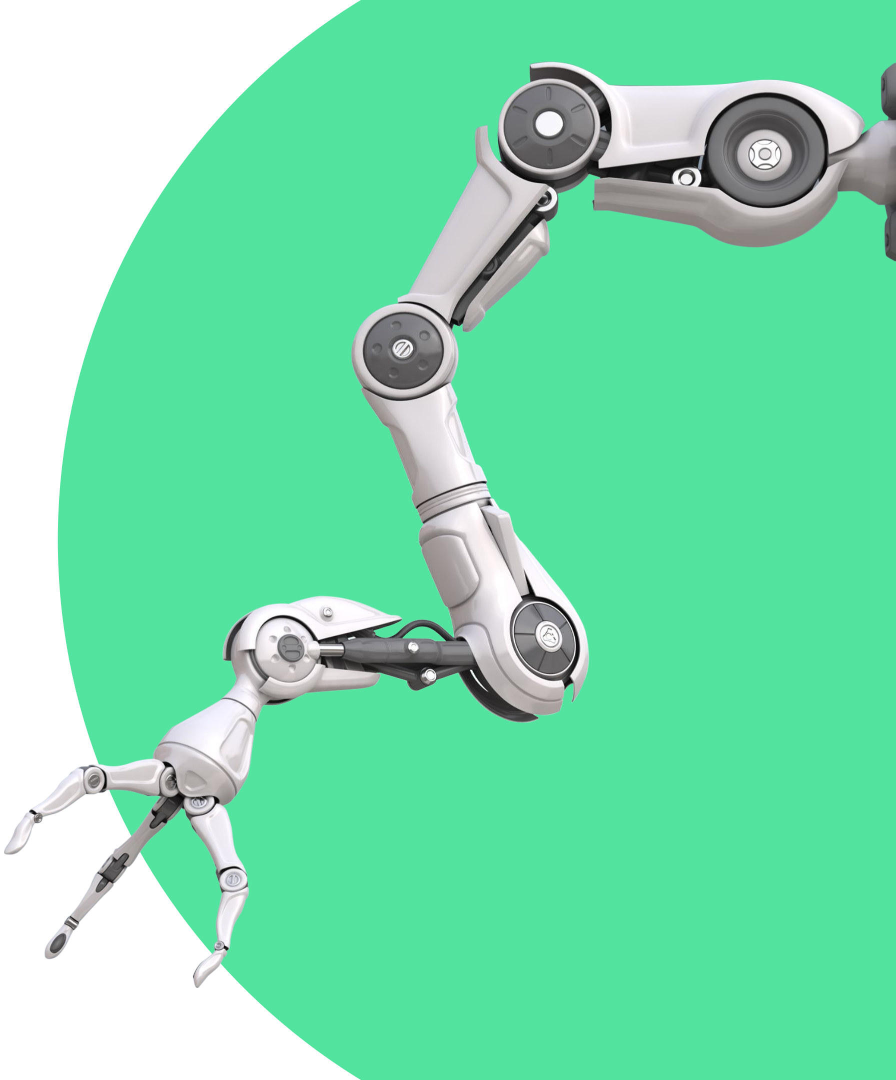
MoveIt2

Realtime Motion Planning

ROS Industrial 2020 Annual Meeting



PICKNIK

Dave Coleman, PhD
CEO, PickNik Robotics
davetcoleman

About PickNik Robotics

We are your partners in strategically developing custom robotics software, while de-risking open source usage.

2015
Incorporated

15
Employees

127 years
Combined Experience

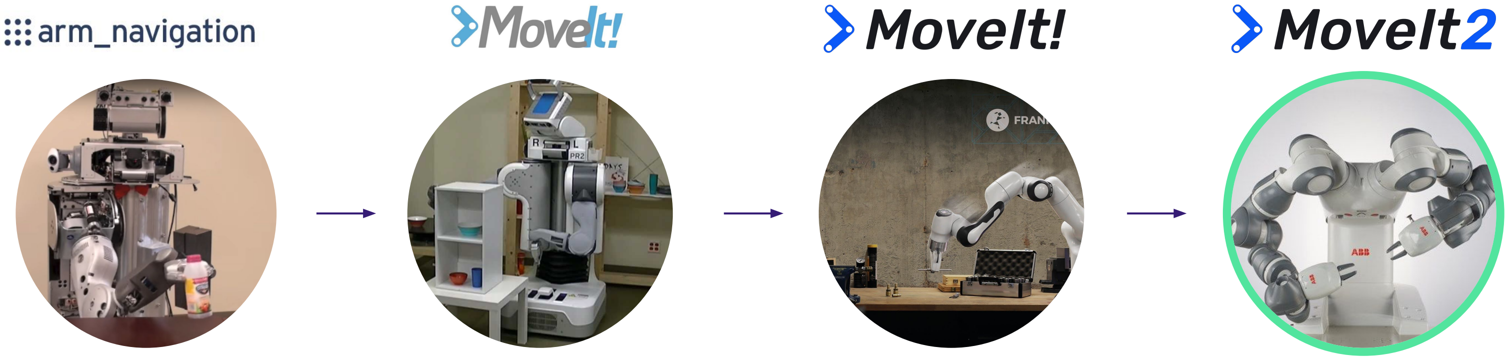
3 Masters
In Robotics

6 PhDs
In Robotics

30+
Client Partners To Date

Headquartered in Boulder, Colorado

MoveIt: A Hardened Motion Planning Platform



arm_navigation → MoveIt! → MoveIt! → MoveIt2

MoveIt Capabilities

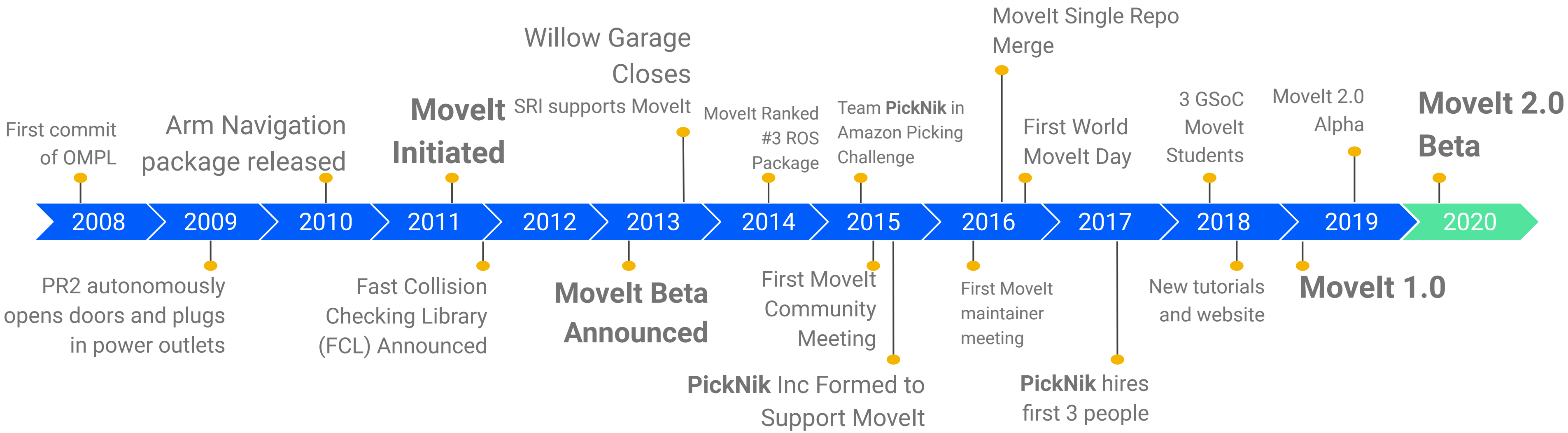
- Motion Planning
    - Generate high-degree of freedom trajectories through cluttered environments and avoid local minimums
- Manipulation
    - Analyze and interact with your environment with grasp generation
- Inverse Kinematics
    - Solve for joint positions for a given pose, even in over-actuated arms
- Control
    - Execute time-parameterized joint trajectories to low level hardware controllers through common interfaces
- 3D Perception
    - Connect to depth sensors and point clouds with Octomaps
- Collision Checking
    - Avoid obstacles using geometric primitives, meshes, or point clouds

MoveIt

109,880	Unique users to moveit.ros.org in 2019
23,662	Downloads per month of moveit_core
542	Academic citations of MoveIt
152	Robot types integrated to work with MoveIt
4200	Members of Discourse, MoveIt's Discussion Forum
509	Github users have starred the MoveIt project
187	Github code contributors to MoveIt
13	International locations participated in World MoveIt Day 2018
310	In-person participants of World MoveIt Day 2018

PICKNIK

MoveIt2
Timeline



2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

2020

First commit of OMPL

Arm Navigation package released

MoveIt Initiated

Willow Garage Closes

SRI supports MoveIt

MoveIt Ranked #3 ROS Package

Team **PickNik** in Amazon Picking Challenge

MoveIt Single Repo Merge

First World MoveIt Day

3 GSoC MoveIt Students

MoveIt 2.0 Alpha

MoveIt 2.0 Beta

PR2 autonomously opens doors and plugs in power outlets

Fast Collision Checking Library (FCL) Announced

MoveIt Beta Announced

First MoveIt Community Meeting

PickNik Inc Formed to Support MoveIt

First MoveIt maintainer meeting

PickNik hires first 3 people

New tutorials and website

MoveIt 1.0

A Feature-Rich Ecosystem

PICKNIK

Global Planners

- OMPL
- SBPL
- TrajOpt
- STOMP
- CHOMP

Cartesian Planners

- RobotState
- Descartes
- JogArm
- PilzIndustrial Motion

Inverse Kinematics

- KDL
- IKFast
- TrackIK
- LMA
- BioIK

Grasping Libraries

- MoveIt Grasps
- Grasp Pose Detection (GPD)
- Intel OpenVino GPD

Collision Checking

- Fast Collision Library (FCL)
- Bullet

Perception / Octomap

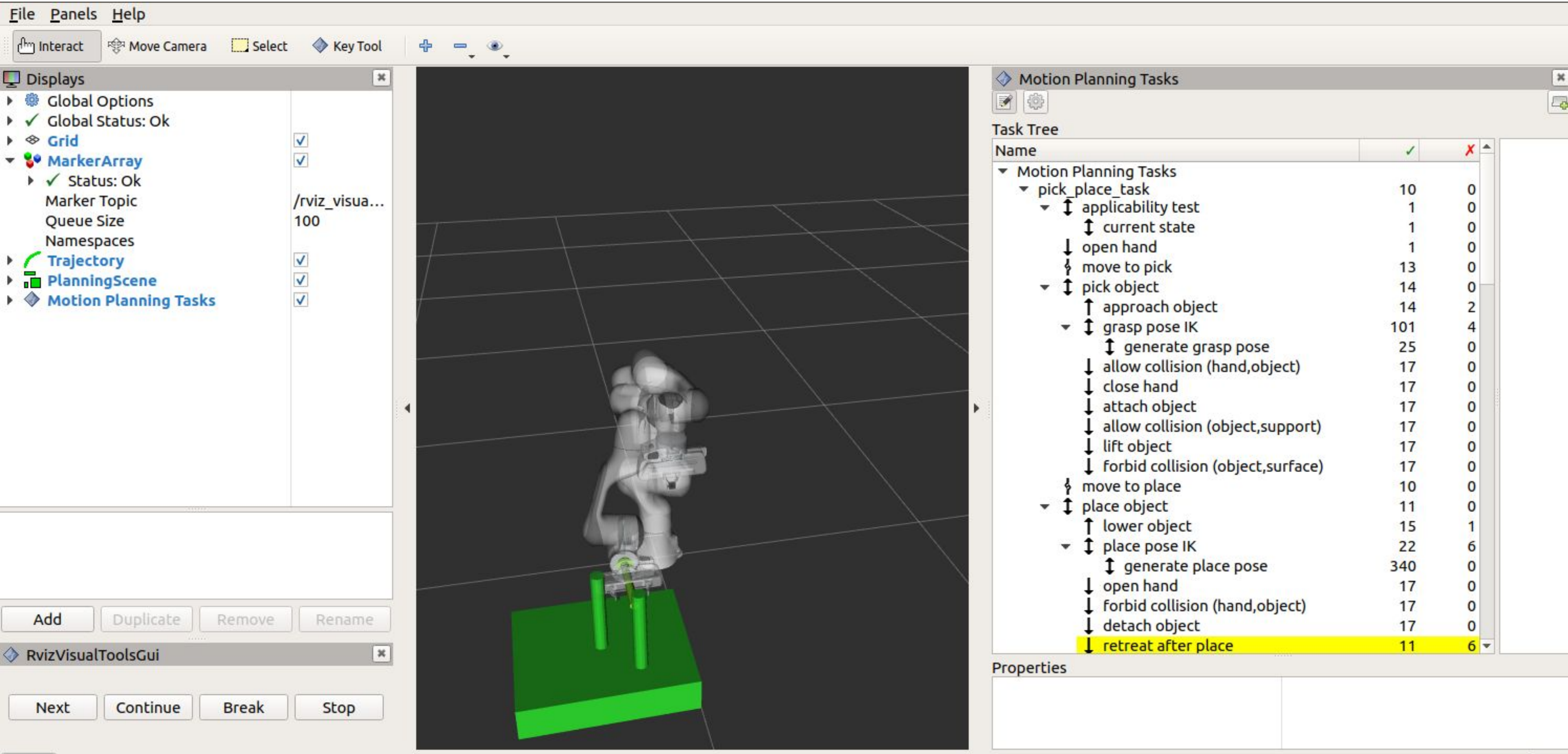
- Depth Images
- Point Clouds

What's new in MoveIt?

Key New Features In MoveIt Ecosystem

- MoveIt Task Constructor
  - *Task Planning*
  - *Robert Haschke, Michael Görner*
- MoveIt Grasps
  - *Geometric-based grasp generation*
  - *Mike Lautman, Dave Coleman*
- MoveIt Cpp
  - *Advanced API for performance*
  - *Henning Kayser*
- MoveIt JogArm
  - *Realtime teleoperation planner*
  - *Andy Zelenak*
- Iterative Cubic Spline Algorithm
  - *Smoother trajectory generation*
  - *Ken Anderson*
- Time-Optimal Trajectory Parameterization
  - *Follow path within bounds on accelerations & velocities*
  - *Michael Ferguson, Henning Kaiser*
- Named Frames on Collision Objects
  - *Subframes for placing objects*
  - *Felix von Drigalski*

MoveIt Task Constructor

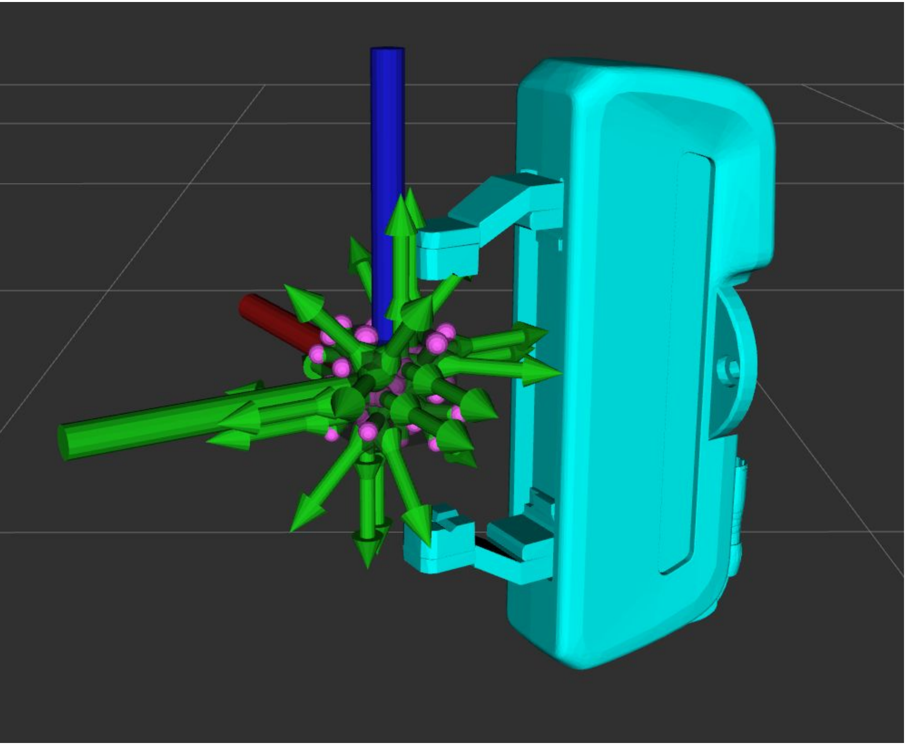


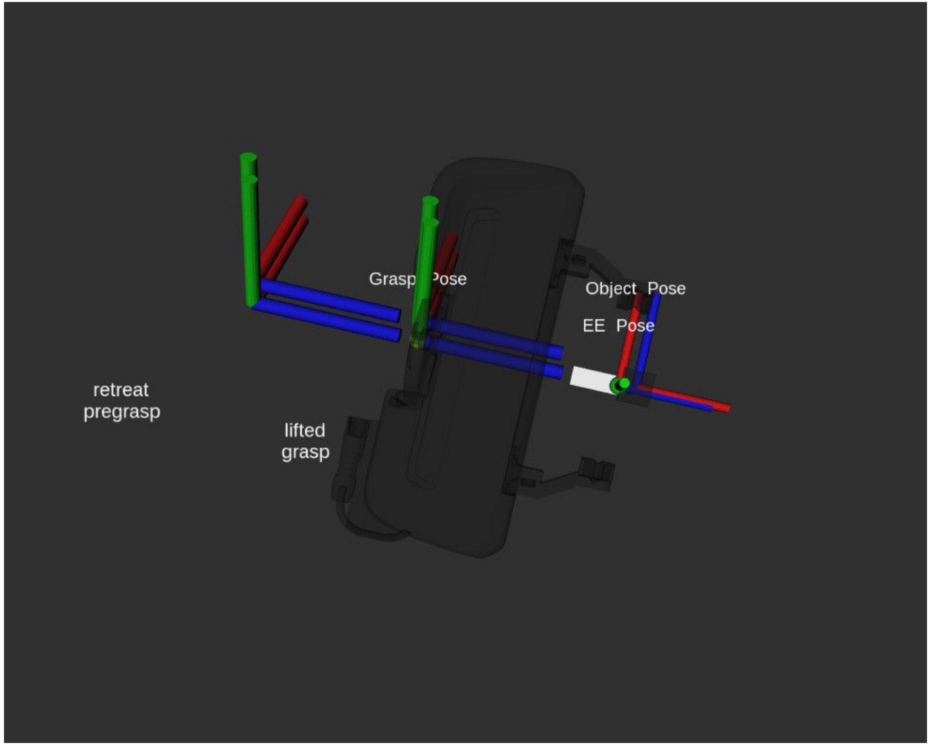
Marker Topic /rviz_visua...
Queue Size 100
Namespaces

Task Tree

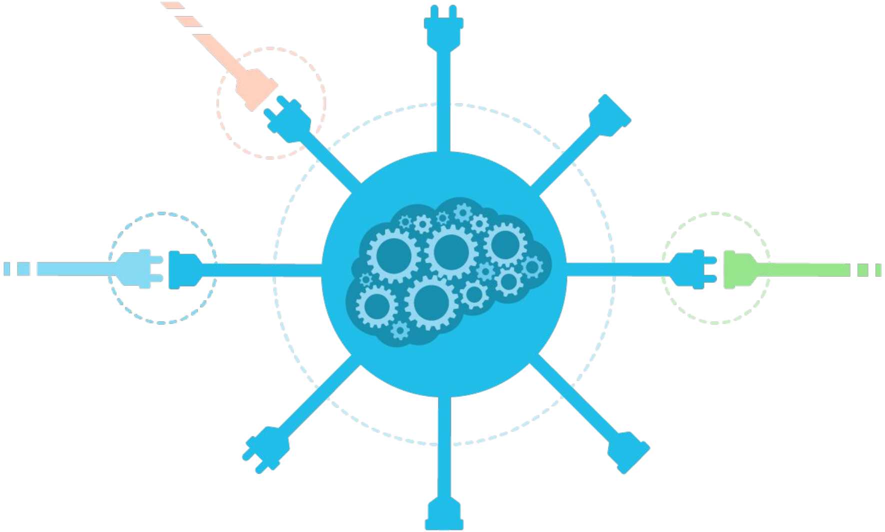
Name	✓	✗
Motion Planning Tasks		
pick_place_task	10	0
applicability test	1	0
current state	1	0
open hand	1	0
move to pick	13	0
pick object	14	0
approach object	14	2
grasp pose IK	101	4
generate grasp pose	25	0
allow collision (hand,object)	17	0
close hand	17	0
attach object	17	0
allow collision (object,support)	17	0
lift object	17	0
forbid collision (object,surface)	17	0
move to place	10	0
place object	11	0
lower object	15	1
place pose IK	22	6
generate place pose	340	0
open hand	17	0
forbid collision (hand,object)	17	0
detach object	17	0
retreat after place	11	6

MoveIt Grasps





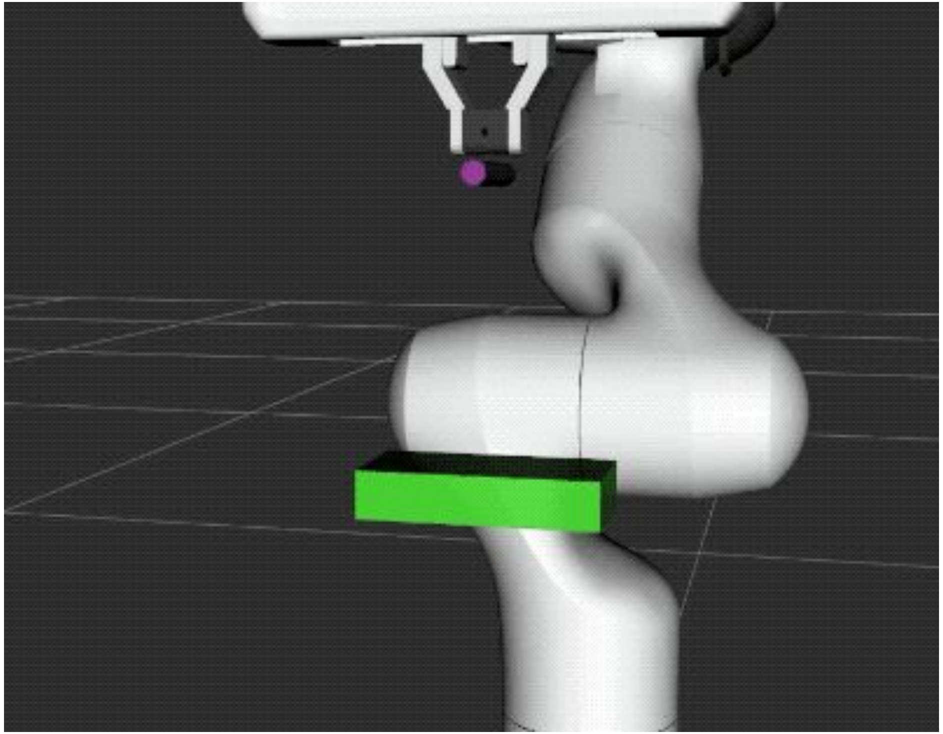
MoveIt Cpp Interface



- Designed by Industry-requested needs
- Speeds up manipulation product development
- As simple as current MoveGroup
- Disables ROS 1 performance bottleneck
- Direct access to core components provided as needed
- Multi-robot support

MoveIt JogArm

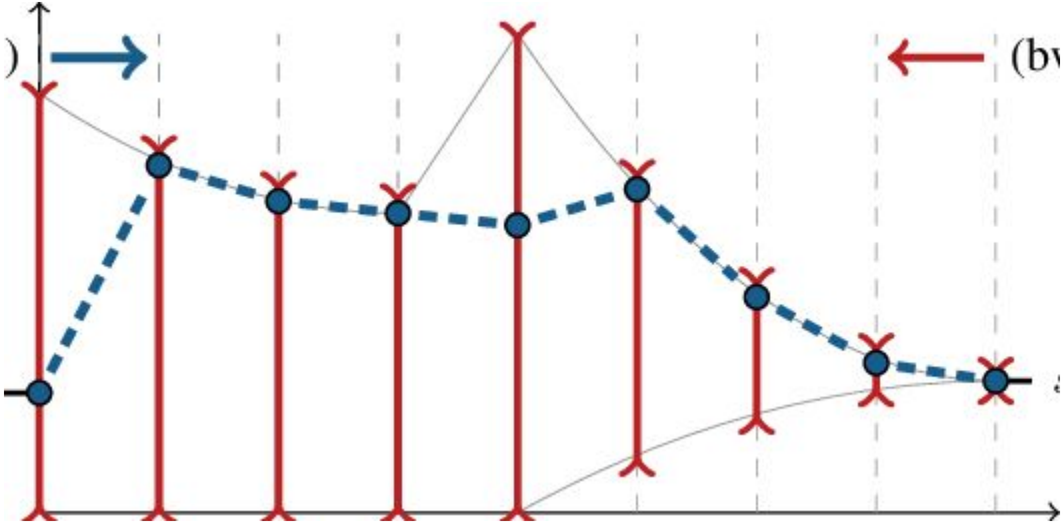
Named Frames on Collision Objects



Time Parameterization

- Iterative Cubic Spline Algorithm
    - *Smoother trajectory generation*
    - *Ken Anderson*

- Time-Optimal Trajectory Parameterization
    - *Follow path within bounds on accelerations & velocities*
    - *Michael Ferguson, Henning Kaiser*



ROS 2 & Realtime

Why ROS 2?

- Realtime support possible
- Multi-platform support: Linux, Windows, OSX
- Production-ready framework based on industry feedback of ROS 1
- DDS: open communication standard

Why care about realtime?

- Vital to many robotics systems, particularly safety and mission critical apps
  - Autonomous vehicles, spacecrafts, and industrial manufacturing.

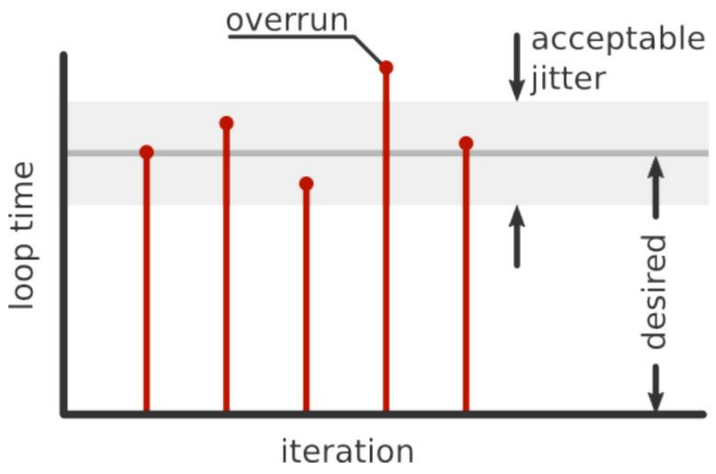
2 primary types of Realtime:

- **Hard realtime** - missing a deadline is considered a system failure
  - Safety- or mission-critical systems
  - Reactor, aircraft and spacecraft control
- **Soft realtime** - missing a deadline has a cost, but is not catastrophic
  - Reduced quality of service
  - Audio / video streaming and playback

Realtime Computing

Determinism, not performance

- Correct computation guaranteed to be delivered within fixed time allotment
- Failure to respond is as bad as a wrong response



Credits to Jackie Kay's 2015 ROSCon presentation

Applying Realtime: Best Practices

- Realtime Operating System (RTOS)
  - Linux + RT Preempt (soft realtime)
  - Xenomai (hard realtime)
- Zero memory copy message passing:
  - Shared memory between threads or processes
- Lock-free circular buffers
- Prioritize real-time threads
- Avoid system calls (memory allocations, printing to console, mutexes)

Note these techniques have largely been available in ROS 1, e.g. MoveItCpp.

Types of Middleware Communication

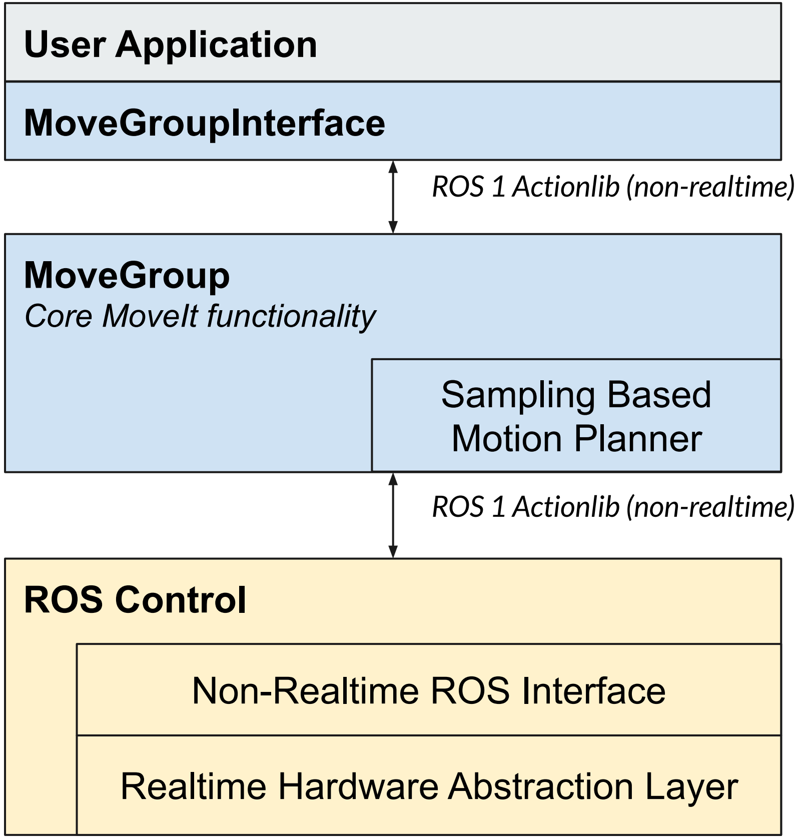
- Inter-process
  - DDS can deliver soft realtime comms
  - Customizable QoS, can be tuned for real-time use-case
- Intra-process (several options)
  - Efficient (zero-copy) shared pointer transport
  - Shared memory with read-only and write-only partition
  - Non-locking circular message queues
- Same-thread
  - No need for synchronization primitives. Simple, fast

Realtime Motion Planning

- Enables:
  - Closed loop, reactive control
  - Streaming joint commands (torques, velocities) to robot arms at high rates (e.g. >1000 Hz)
- Improves:
  - Reliability
  - Extended uptime

MoveIt 1.0

Out of Box Approach



User Application

MoveGroupInterface

ROS 1 Actionlib (non-realtime)

MoveGroup
Core MoveIt functionality

Sampling Based Motion Planner

ROS 1 Actionlib (non-realtime)

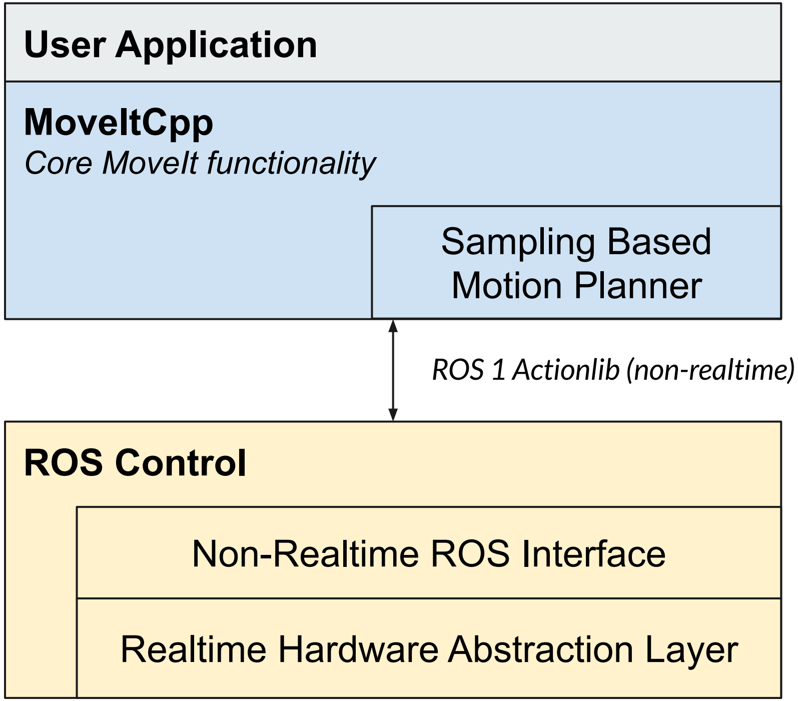
ROS Control

Non-Realtime ROS Interface

Realtime Hardware Abstraction Layer

MoveIt 1.0

New Advanced Approach with MoveItCpp



User Application

MoveItCpp
Core MoveIt functionality

Sampling Based Motion Planner

ROS 1 Actionlib (non-realtime)

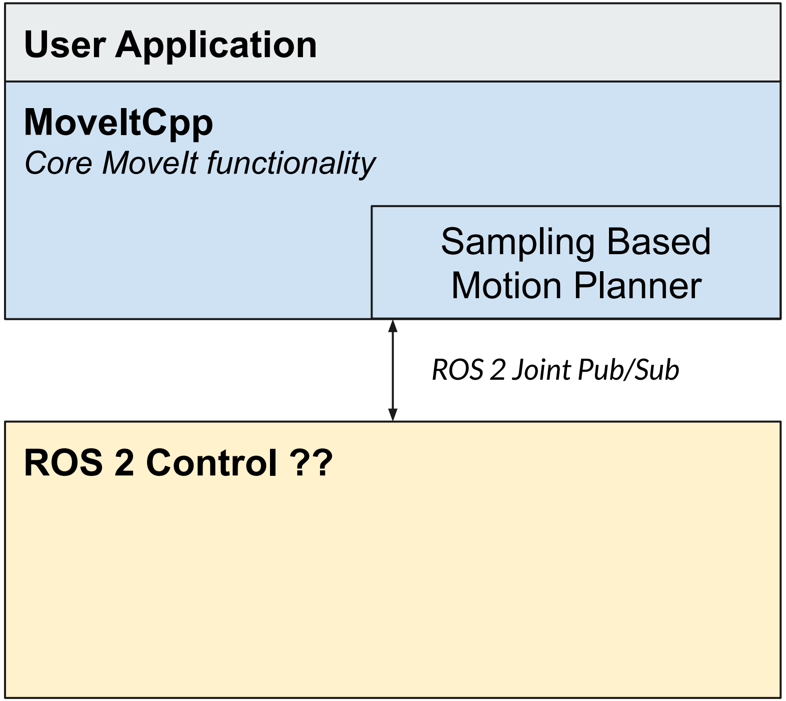
ROS Control

Non-Realtime ROS Interface

Realtime Hardware Abstraction Layer

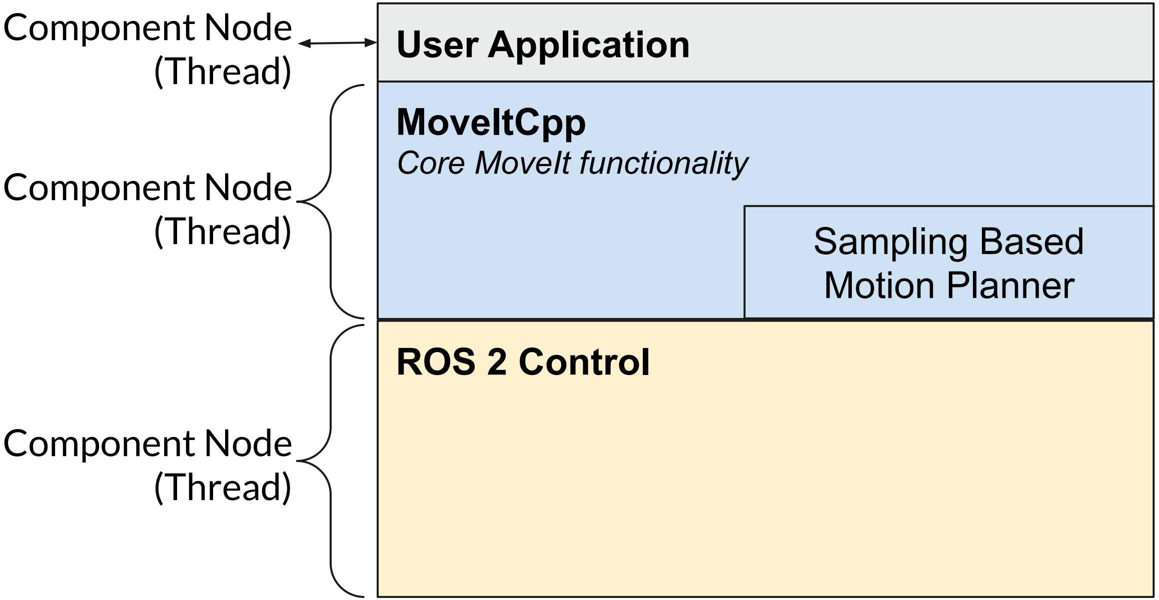
MoveIt 2.0

Current Beta Implementation



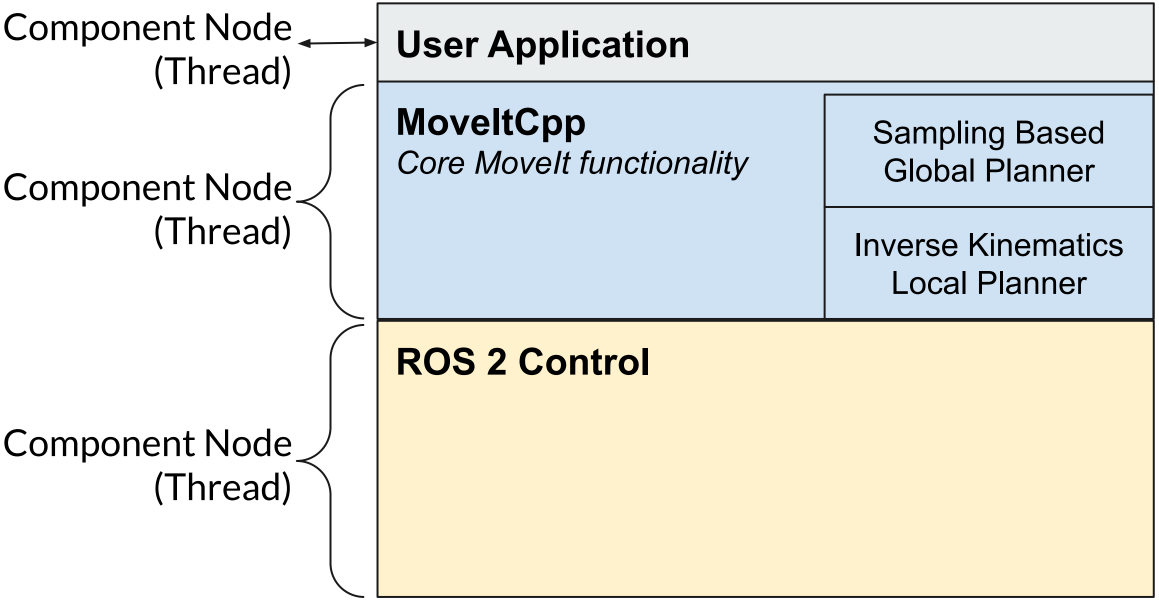
MoveIt 2.0

Proposed Implementation with ROS 2 Component Nodes



MoveIt 2.0

Hybrid Motion Planning



Global vs Local Planning

Global Planning (assuming sampling)

- Pros:
  - Plan around complex obstacles
  - Avoid getting stuck in local minimum
  - Complete: will find solution if exists
- Cons:
  - Slower computation time
  - Not realtime
  - Not deterministic

Local Planning (assuming jacobian based)

- Pros:
  - Fast / Reactive
  - Deterministic
  - Well suited for visual servoing
- Cons:
  - Gets stuck in local minimum
  - Fewer collision safety guarantees

Hybrid Planning

- Simultaneously plan globally and locally
- Plan at different speeds in separate thread:
    - Global planner (full collision checking): ~30Hz
    - Local Planner (IK-based, field-based): ~300Hz

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

Roadmap

Roadmap

MoveIt 2.0 Releases

- Alpha
  - Released June 2019
  - ROS 2 Dashing Diademata
- Beta
  - Released February 2020
  - ROS 2 Eloquent Elusor

MoveIt2

Milestone 1

Straight Port to ROS 2

- Fully migrate existing MoveIt packages to ROS 2
  - Wrap up Acutronic's work porting core MoveIt functionality
- Leverage ROS 2:
  - Build system (ament), middleware, logging, parameters
- Cleanup MoveIt 2 codebase

YOU ARE HERE

Milestone 2

Realtime Support

- Reactive, closed-loop control to sensor input
  - Visual servoing, octomap updates
  - Preempt motion if new collision detected
- Seperate 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

MoveIt Survey Results

91% most excited about ROS 2 realtime control

55% reactive planning and closed loop control

48% better integration with lower level realtime control

48% planning with dynamics

Milestone 3

Fully Leverage ROS 2

- Lifecycle management of MoveIt nodes
  - Deterministic startup, reset, & shutdown sequences
- Leverage ROS2 component nodes
  - Ability to run MoveIt as single or multi-process
  - Replace pluginlib with components
- Cleanup API
  - More generic and standalone interfaces

MoveIt Survey Results

47% excited about component nodes

Future Milestones

Determinism

- Out of box / default planners return reliable paths
  - Tune or replace OMPL, BIT*
- Further optimize / smooth paths
  - Default use TOTG, TOPP time parameterization
  - Use post-processing optimization (STOMP, TrajOpt)
- Fully featured Cartesian Planner
  - Like Descartes but better and fully integrated
- Force-torque control

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

Machine Learning

- Neural-network based motion planning - new plugins
  - General near-optimal heuristics for path planning
  - e.g. MPNet

Progress on Roadmap

1. **Finish migration of MoveIt 1 packages**
2. Document how to use ROS1 bridges for legacy support
3. Merge and simplify ecosystem repositories
4. Address realtime support
5. Improve deterministic planning

Getting Involved

Upcoming Events

Google Summer of Code

MoveItCon

November 17th

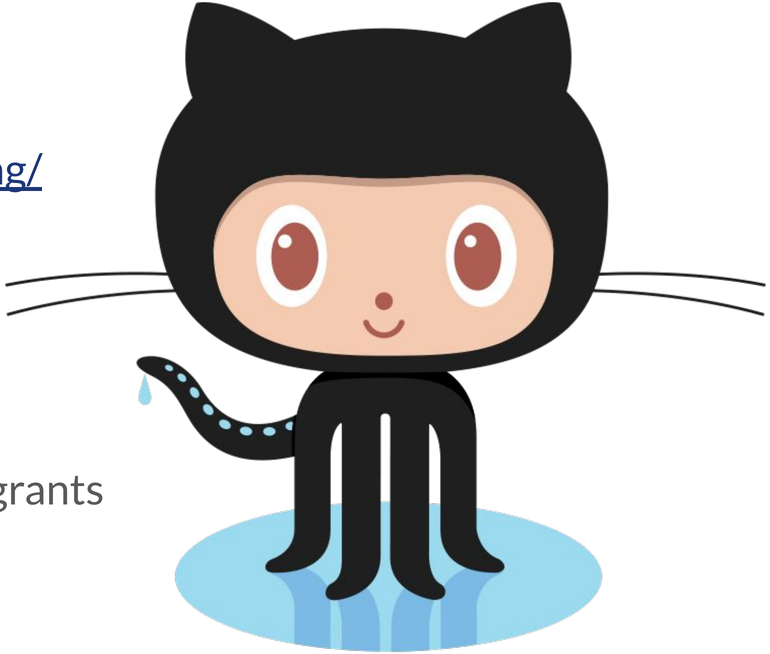
WorldMoveItDay

Contributing to MoveIt

https://moveit.ros.org/documentation/contributing/

Many approaches:

- Adding New Features
- Helping with MoveIt 2 Port
- Financial contributions via code sprints and grants
- Enhancing Documentation
- Reporting & Fixing Bugs



Thanks!

Dave Coleman
dave@picknik.ai
🐦 @waffle

PickNik Robotics
(+1) 720 513 2221
1942 Broadway, Suite 314
Boulder, Colorado 80302
🐦 @picknikrobotics