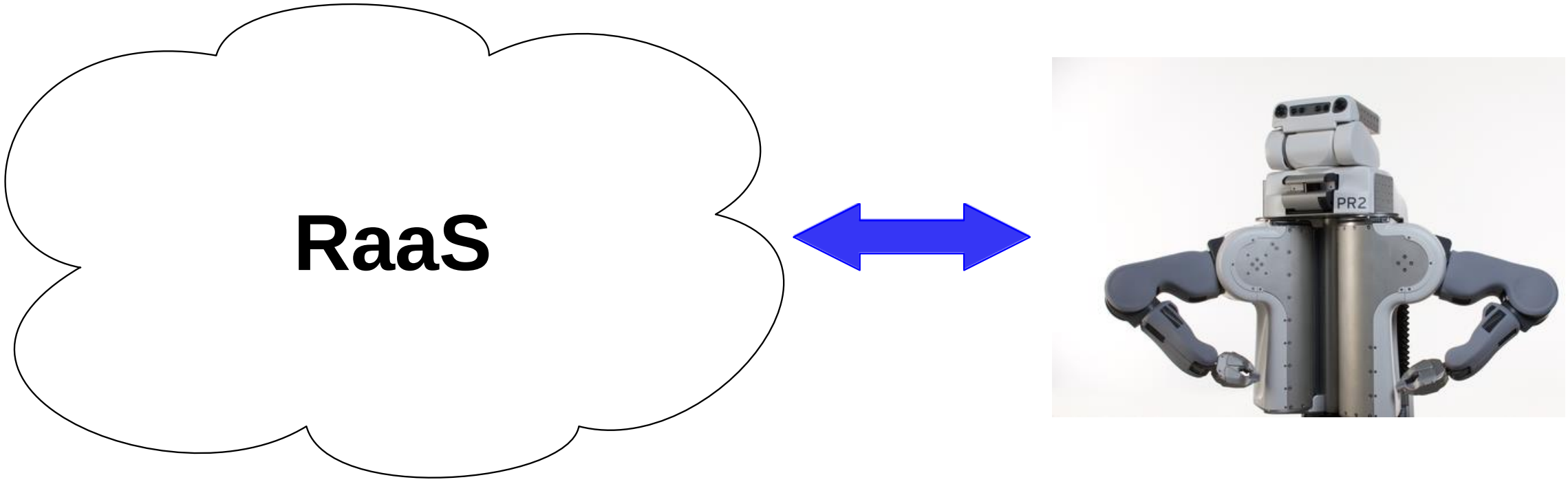
# RaaS: Robotics as a Service

## A Service Model for Cloud Robotics

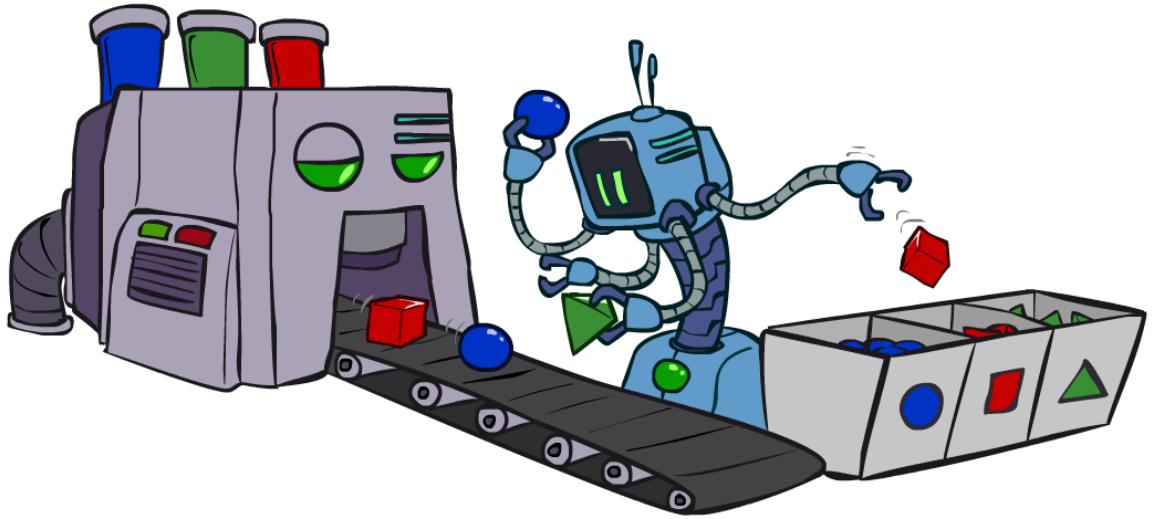



Arjun Singh, Ben Kehoe, Ken Goldberg, Pieter Abbeel

University of California, Berkeley

# Example Use Case

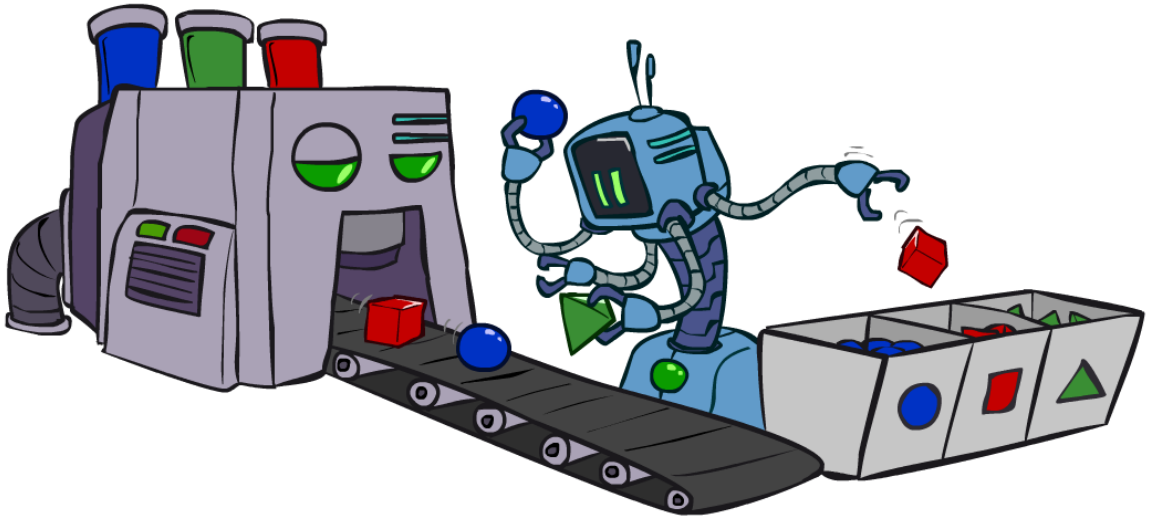
- Objects coming down conveyer belt
- Identify each object, pick it up, and place into correct bin
  - Low-level control
  - Object recognition
  - Grasp planning
  - Motion planning



Drawing: Ketrina Yim

# How to Solve It with ROS

- Acquire computing hardware
- Install ROS and relevant packages
- Execute and maintain software and computing hardware



Drawing: Ketrina Yim

# Pros and Cons of ROS

- Huge positive impact on robotics (especially sharing)
- Core great for messaging, low-level control – components that make a robot *operating system*
- Asking it to do too much
  - All code for everything robotics related

```
arjun@perception-data ~ sudo apt-get install ros-groovy-ros-base
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  binfmt-support comerr-dev docutils-common docutils-doc krb5-multidev libapr1 libapr1-dev libaprutil1 libaprutil1-dev libbz2-dev libgssrpc4 libgtest-dev libkadm5clnt-mit8
  libkadm5srv-mit8 libkdb5-6 libkrb5-dev libldap-2.4-2 libldap2-dev liblog4cxx10 liblog4cxx10-dev libmysqlclient-dev libmysqlclient18 libpq-dev libpq5 libsqlite3-dev
  libtinyxml-dev libtinyxml2.6.2 libyaml-0-2 mysql-common python-catkin-pkg python-docutils python-empy python-lxml python-netifaces python-nose python-paramiko python-pygments
  python-roman python-rosdep python-rosdistro python-rospkg python-yaml ros-groovy-catkin ros-groovy-cpp-common ros-groovy-gencpp ros-groovy-genlisp ros-groovy-genmsg
  ros-groovy-genpy ros-groovy-message-filters ros-groovy-message-generation ros-groovy-message-runtime ros-groovy-mk ros-groovy-ros ros-groovy-ros-comm ros-groovy-rosbag
  ros-groovy-rosbash ros-groovy-rosboost-cfg ros-groovy-rosbuild ros-groovy-rosclean ros-groovy-rosconsole ros-groovy-roscpp ros-groovy-roscpp-serialization
  ros-groovy-roscpp-traits ros-groovy-roscreate ros-groovy-rosgraph ros-groovy-rosgraph-msgs ros-groovy-roslang ros-groovy-roslaunch ros-groovy-roslib ros-groovy-roslisp
  ros-groovy-rosmake ros-groovy-rosmaster ros-groovy-rosmsg ros-groovy-rosnode ros-groovy-rosout ros-groovy-rospack ros-groovy-rosparam ros-groovy-rospy ros-groovy-rosservice
  ros-groovy-rostest ros-groovy-rostime ros-groovy-rostopic ros-groovy-rosunit ros-groovy-roswtf ros-groovy-std-msgs ros-groovy-std-srvs ros-groovy-topic-tools
  ros-groovy-xmlrpcpp sbcl uuid-dev
```

Packages required for barebones ROS installation

```
arjun@perception-data ~ sudo apt-get install ros-groovy-desktop-full
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  autoconf automake autotools-dev binfmt-support blt collada-dom-dev collada-dom2.4-dp-base collada-dom2.4-dp-dev comerr-dev docgenerator docutils-common docutils-doc ffmpeg
  generatorrunner graphviz hddtemp krb5-multidev libapiextractor-dev libapiextractor0.10 libapr1 libapr1-dev libaprutil1 libaprutil1-dev libasound2 libasound2-dev libassimp-dev
  libassimp2 libav-tools libavahi-client-dev libavahi-common-dev libavdevice53 libavfilter2 libbz2-dev libcaca-dev libcdt4 libcgraph5 libcppunit-1.12-1 libcppunit-dev libcurl3
  libcurl4-openssl-dev libdb4.8 libdbus-1-3 libdbus-1-dev libeigen3-dev libfltk1.1 libfltk1.1-dev libfreeimage-dev libfreeimage3 libgcrypt11 libgcrypt11-dev libgenrunner-dev
  libgenrunner0.6 libgl2ps-dev libgl2ps0 libgnutls-dev libgnutls-openssl27 libgnutls26 libgnutlsxx27 libgpg-error-dev libgraph4 libgssrpc4 libgtest-dev libgvc5 libgvpr1
  libidn11-dev libjs-sphinxdoc libjs-underscore libkadm5clnt-mit8 libkadm5srv-mit8 libkdb5-6 libkrb5-dev libldap-2.4-2 libldap2-dev liblog4cxx10 liblog4cxx10-dev libltdl-dev
  libmysqlclient-dev libmysqlclient18 libncurses5-dev libnetcdf-dev libnetcdf6 libnetpbm10 libnetpbm10-dev libodbc1 libogg-dev libogre-1.7.4 libogre-dev libopenjpeg2
  libp11-kit-dev libpathplan4 libphonon4 libpoco-dev libpococrypto9 libpocodata9 libpocofoundation9 libpocomysql9 libpoconet9 libpoconetssl9 libpocoodbc9 libpocosqlite9
  libpocoutil9 libpocoxml9 libpocozip9 libpostproc52 libpq-dev libpq5 libprotobuf-dev libprotobuf-lite7 libprotoc-dev libpulse-dev libpulse-mainloop-glib0 libpulse0
  libpyside-dev libpyside-py3-1.1 libpyside1.1 libpython3.2 libqhull-dev libqhull5 libqt4-dbus libqt4-declarative libqt4-designer libqt4-dev libqt4-help libqt4-network
  libqt4-opengl libqt4-opengl-dev libqt4-qt3support libqt4-script libqt4-scripttools libqt4-sql libqt4-sql-sqlite libqt4-svg libqt4-test libqt4-xml libqt4-xmlpatterns
  libqtassistantclient4 libqtcore4 libqtgui4 libqtwebkit-dev libqtwebkit4 libqwt-dev libqwt5-qt4 libqwt6 librtmp-dev libsdl-image1.2 libsdl-image1.2-dev libsdl1.2-dev
  libshiboken-dev libshiboken-py3-1.1 libshiboken1.1 libslang2-dev libsqlite3-dev libsvn1 libtar-dev libtar0 libtasn1-3-dev libtbb-dev libtbb2 libtheora-dev libtinfo-dev
  libtinyxml-dev libtinyxml2.6.2 libtool libv4l-dev libvtk5-dev libvtk5.8 libwxbase2.8-0 libwxgtk2.8-0 libxaw7-dev libxml2 libxml2-dev libxmu-dev libxmu-headers libxpm-dev
  libxss-dev libxss1 libyaml-0-2 libyaml-dev libzzip-0-13 libzzip-dev m4 mysql-common phonon phonon-backend-gstreamer python-catkin-pkg python-docutils python-empy python-epydoc
  python-glade2 python-jinja2 python-lxml python-matplotlib python-matplotlib-data python-netifaces python-nose python-opengl python-paramiko python-pydot python-pygments
  python-pyparsing python-pyside python-pyside.phonon python-pyside.qtcore python-pyside.qtdeclarative python-pyside.qtgui python-pyside.qthelp python-pyside.qtnetwork
  python-pyside.qtopengl python-pyside.qtscript python-pyside.qtsql python-pyside.qtsvg python-pyside.qttest python-pyside.qtuitools python-pyside.qtwebkit python-pyside.qtxml
  python-qt4 python-qt4-dev python-qt4-gl python-qwt5-qt4 python-roman python-rosdep python-rosdistro python-rospkg python-sip python-sip-dev python-sphinx python-tk python-tz
  python-wxgtk2.8 python-wxversion python-yaml qdbus qt4-linguist-tools qt4-qmake ros-groovy-actionlib ros-groovy-actionlib-msgs ros-groovy-angles ros-groovy-bfl ros-groovy-bond
  ros-groovy-bond-core ros-groovy-bondcpp ros-groovy-bondpy ros-groovy-bullet ros-groovy-camera-calibration ros-groovy-camera-calibration-parsers ros-groovy-camera-info-manager
  ros-groovy-catkin ros-groovy-class-loader ros-groovy-collada-parser ros-groovy-collada-urdf ros-groovy-common ros-groovy-common-msgs ros-groovy-common-rosdeps
  ros-groovy-common-tutorials ros-groovy-compressed-depth-image-transport ros-groovy-compressed-image-transport ros-groovy-console-bridge ros-groovy-convex-decomposition
  ros-groovy-cpp-common ros-groovy-cv-bridge ros-groovy-depth-image-proc ros-groovy-diagnostic-aggregator ros-groovy-diagnostic-analysis ros-groovy-diagnostic-common-diagnostics
  ros-groovy-diagnostic-msgs ros-groovy-diagnostic-updater ros-groovy-diagnostics ros-groovy-documentation ros-groovy-driver-base ros-groovy-driver-common
  ros-groovy-dynamic-reconfigure ros-groovy-eigen-conversions ros-groovy-executive-smach ros-groovy-filters ros-groovy-flann ros-groovy-gencpp ros-groovy-genlisp
  ros-groovy-genmsg ros-groovy-genpy ros-groovy-geometry ros-groovy-geometry-experimental ros-groovy-geometry-msgs ros-groovy-geometry-tutorials ros-groovy-image-common
  ros-groovy-image-geometry ros-groovy-image-pipeline ros-groovy-image-proc ros-groovy-image-rotate ros-groovy-image-transport ros-groovy-image-transport-plugins
  ros-groovy-image-view ros-groovy-interactive-markers ros-groovy-ivcon ros-groovy-kdl-conversions ros-groovy-kdl-parser ros-groovy-laser-assembler ros-groovy-laser-filters
  ros-groovy-laser-geometry ros-groovy-laser-pipeline ros-groovy-message-filters ros-groovy-message-generation ros-groovy-message-runtime ros-groovy-mk ros-groovy-nav-msgs
  ros-groovy-navigation ros-groovy-navigation-tutorials ros-groovy-nodelet ros-groovy-nodelet-core ros-groovy-nodelet-topic-tools ros-groovy-opencv2 ros-groovy-orocos-kdl
  ros-groovy-pcl ros-groovy-pcl-msgs ros-groovy-pcl-ros ros-groovy-perception-pcl ros-groovy-physics-ode ros-groovy-pluginlib ros-groovy-polled-camera
  ros-groovy-python-qt-binding ros-groovy-qt-dotgraph ros-groovy-qt-gui ros-groovy-qt-gui-app ros-groovy-qt-gui-core ros-groovy-qt-gui-cpp ros-groovy-qt-gui-py-common
  ros-groovy-resource-retriever ros-groovy-robot-model ros-groovy-robot-model-tutorials ros-groovy-robot-model-visualization ros-groovy-robot-state-publisher ros-groovy-ros
  ros-groovy-ros-comm ros-groovy-ros-tutorials ros-groovy-rosbag ros-groovy-rosbash ros-groovy-rosboost-cfg ros-groovy-rosbuild ros-groovy-rosclean ros-groovy-rosconsole
  ros-groovy-rosconsole-bridge ros-groovy-roscpp ros-groovy-roscpp-serialization ros-groovy-roscpp-traits ros-groovy-roscpp-tutorials ros-groovy-roscreate ros-groovy-rosgraph
  ros-groovy-rosgraph-msgs ros-groovy-roslang ros-groovy-roslaunch ros-groovy-roslib ros-groovy-roslisp ros-groovy-rosmake ros-groovy-rosmaster ros-groovy-rosmsg
```

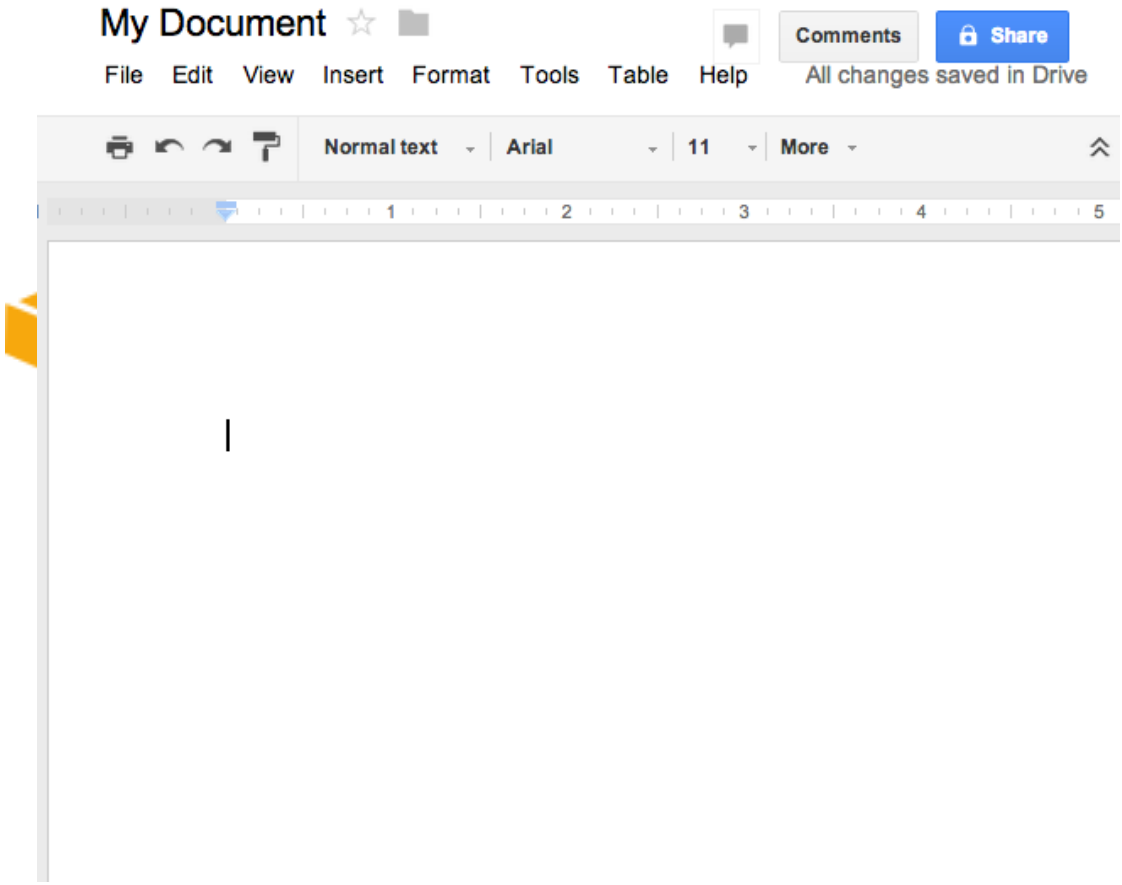
Packages required for full ROS installation

# Pros and Cons of ROS

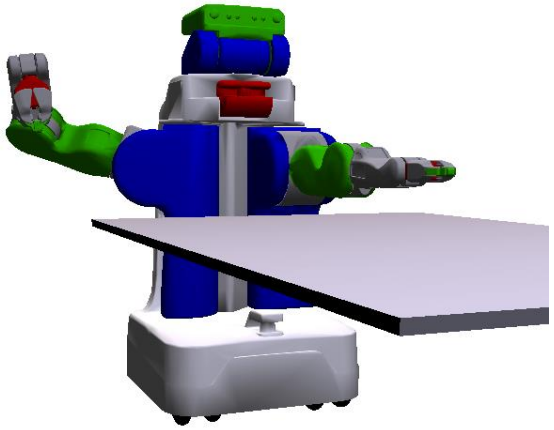
- Intimidating for non-roboticists
- Nontrivial to set up secure distributed networking with ROS -- need to understand VPNs, have control over network environment, etc.
- Dependencies can turn into a nightmare (especially with multiple ROS versions)

# Computing Environment has Changed

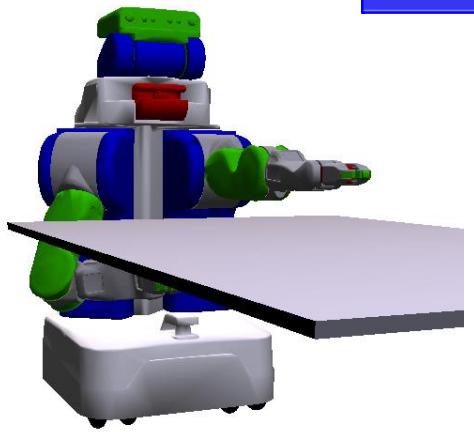
- ROS design started in 2006 – a lot has changed!
- Cloud computing: Easy access to vast numbers of machines
- Software engineering: Service-oriented architectures and Software as a Service
  - E.g. Google Docs vs. Microsoft Office



# Example Services – Motion Planning

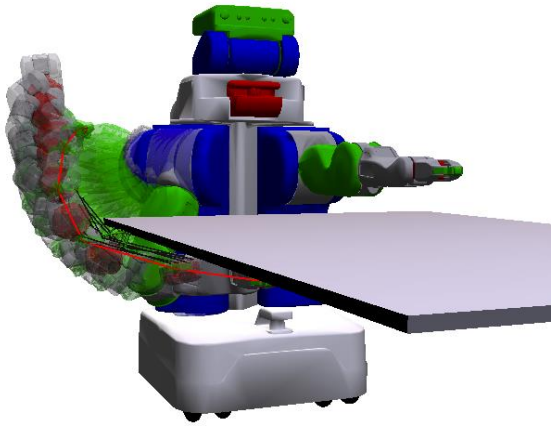


Start State



Goal State

Motion Planning Service

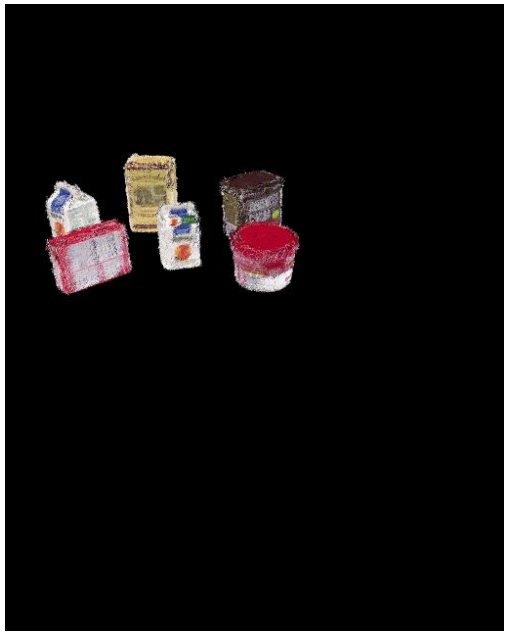


Output Trajectory

# Example Services – Object Recognition

Input Scene

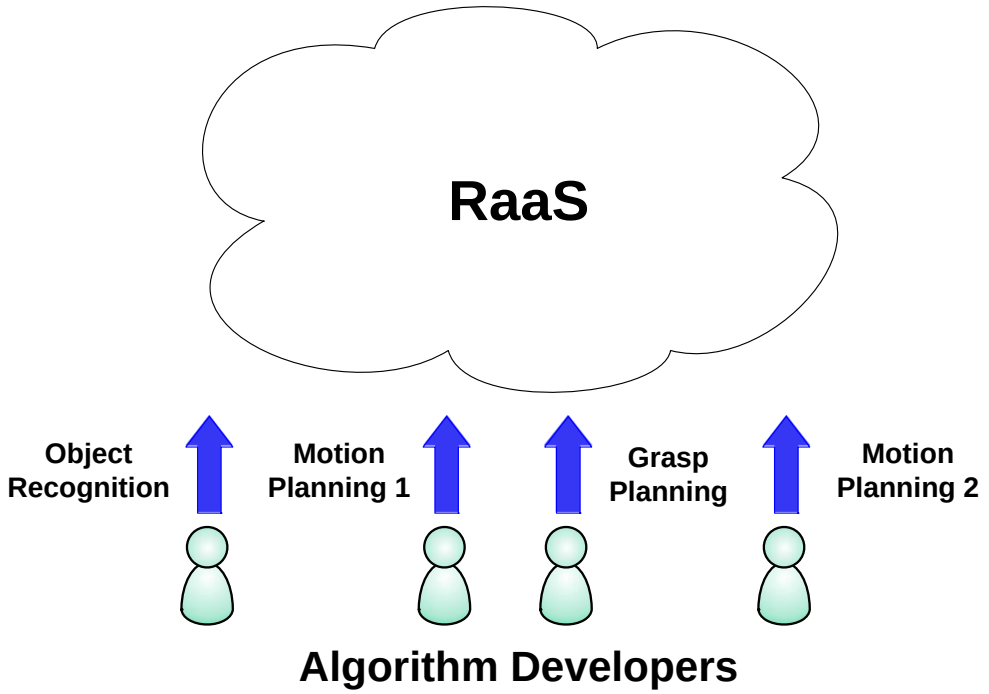
**Object Recognition Service**



Detected Objects and Poses

# Our Idea: Robotics as a Service (RaaS)

- Publish algorithms as services
- Consume services on your robot
- Collaborative data collection




m Users

# Related Work

## Cloud Robotics

| | |
|---|---|
| **Rapyuta (www.rapyuta.org/)** | **Robotics Platform as a Service** |
| Rocon (www.robotconcert.org/) | Higher-level orchestration |
| Goldberg et al. 1995 | Networked robotics |
| Arumugam et al. 2010 | Cloud computing for robots |
| Ciocarlie et al. 2010 | Big data for robotics |
| Kuffner 2010 | Cloud-enabled robots |
| **Remy and Blake 2011** | **Service-oriented robotics** |
| **Blake et al. 2011** | **Service-oriented robotics** |
| **Waibel et al. 2011** | **Shared knowledge** |

## Open-Source Tools

| | |
|---|---|
| MoveIt! (Sucan and Chitta) | Motion planning |
| GraspIt! (Ciocarlie and Miller) | Grasp planning |
| OpenCV (Bradski) | Computer vision |
| OpenRAVE (Diankov) | Motion planning |
| Ladon (www.ladonize.org) | Web service framework |

## Commercial Products

| | |
|---|---|
| Mashape | Index of service APIs |
| PiCloud | Move computation into cloud |

# Why RaaS?

Limited Resources | Code Sharing | Encapsulation | Parallelism | Common Interfaces

# Why RaaS?



- Some robotic platforms have limited onboard computation (e.g. Baxter)
  - Can't run sophisticated robotics algorithms
  - Need offboard computation – use the cloud or buy computers
- Services give a straightforward way to move computation to the cloud

Photo: David Yellen for *IEEE Spectrum*

# Why RaaS?



- Spend months working on your algorithm, finally finish it, and you want to share it
  - Need to figure out what all the dependencies are
  - Need to document how to install everything
  - Need to document how to use your code
  - Don’t have time for any of these
- Packaged service: only worry about API users interact with

# Why RaaS?



- Robotics requires integrating many different components
  - Vision, NLP, control, messaging, planning, grasping, etc. – each alone is complex
  - Huge number of dependencies – something is bound to conflict
- Services force encapsulation of components
  - Use object detection and planning systems without worrying about conflicts

**Object Detection:**
Dependency A 1.0
Dependency B 2.1
Dependency C 2.5

**Motion Planning:**
Dependency B 2.1
Dependency C 3.2

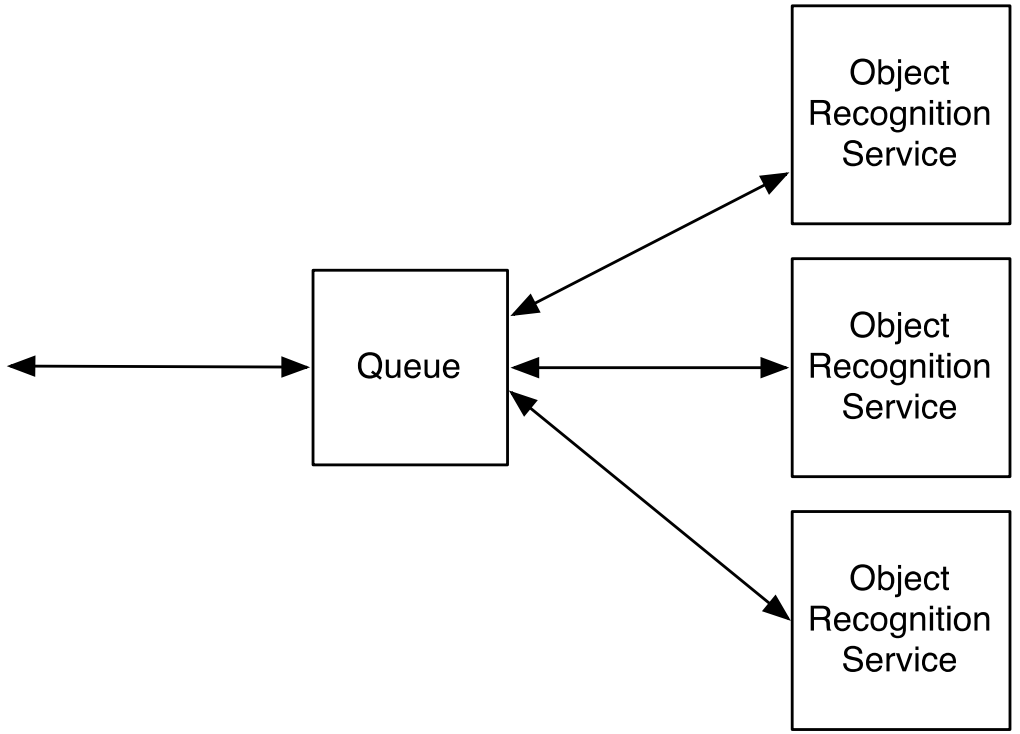
**Object Detection:**
Dependency A 1.0
Dependency B 2.1
Dependency C 2.5

**Motion Planning:**
Dependency B 2.1
Dependency C 3.2

# Why RaaS?



- Service: natural interface for parallelizing computation
- Insulates user from managing parallelism
- Automatically run multiple instances of a service on multiple machines

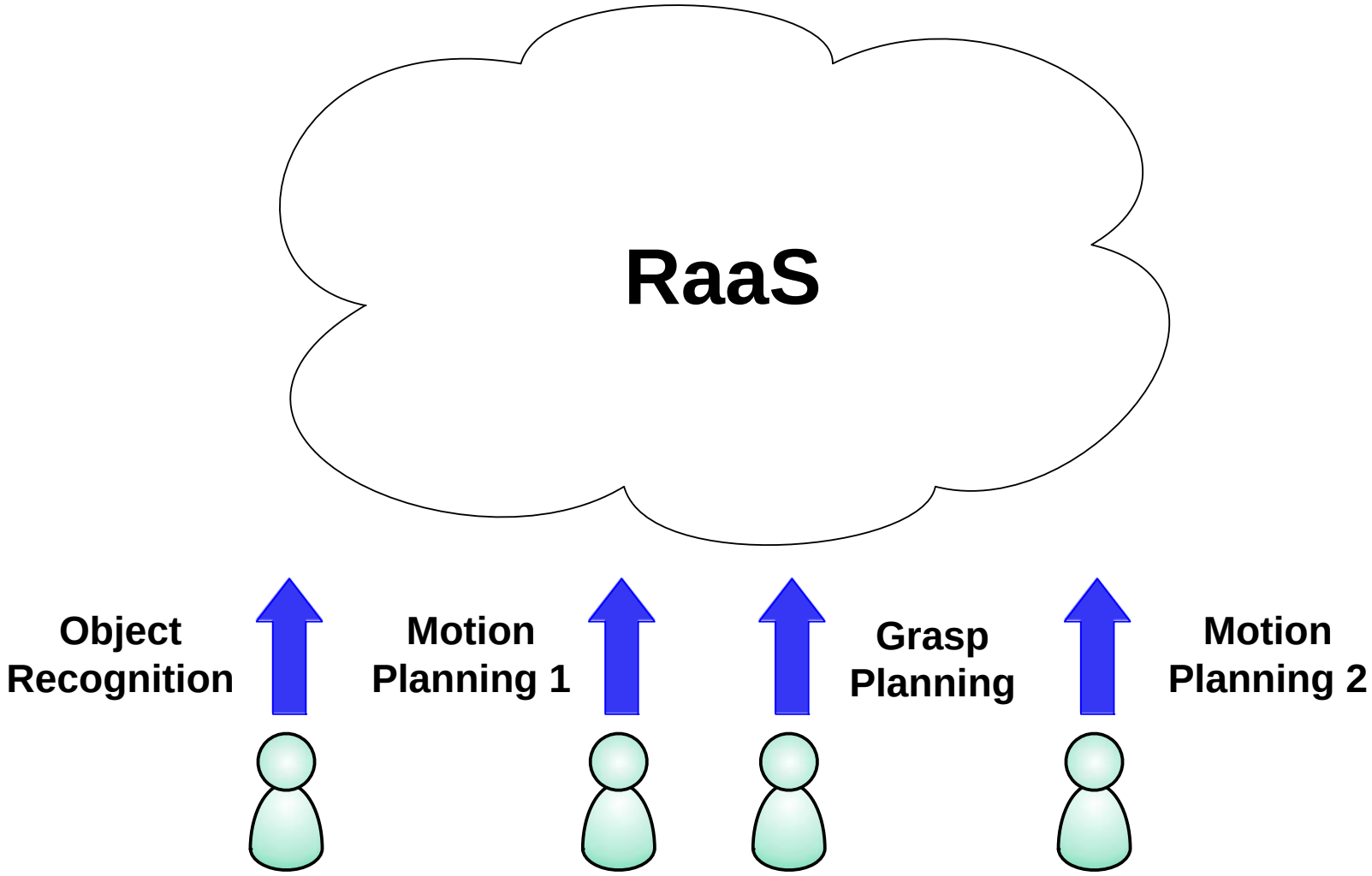
# Why RaaS?



- Easy benchmarking and comparison
- Use common interfaces when defining services
  - E.g. object recognition systems
    - Input: Image
    - Output: List of object identities
    - Can’t expect researchers to implement interfaces at a library level
- Swap out services

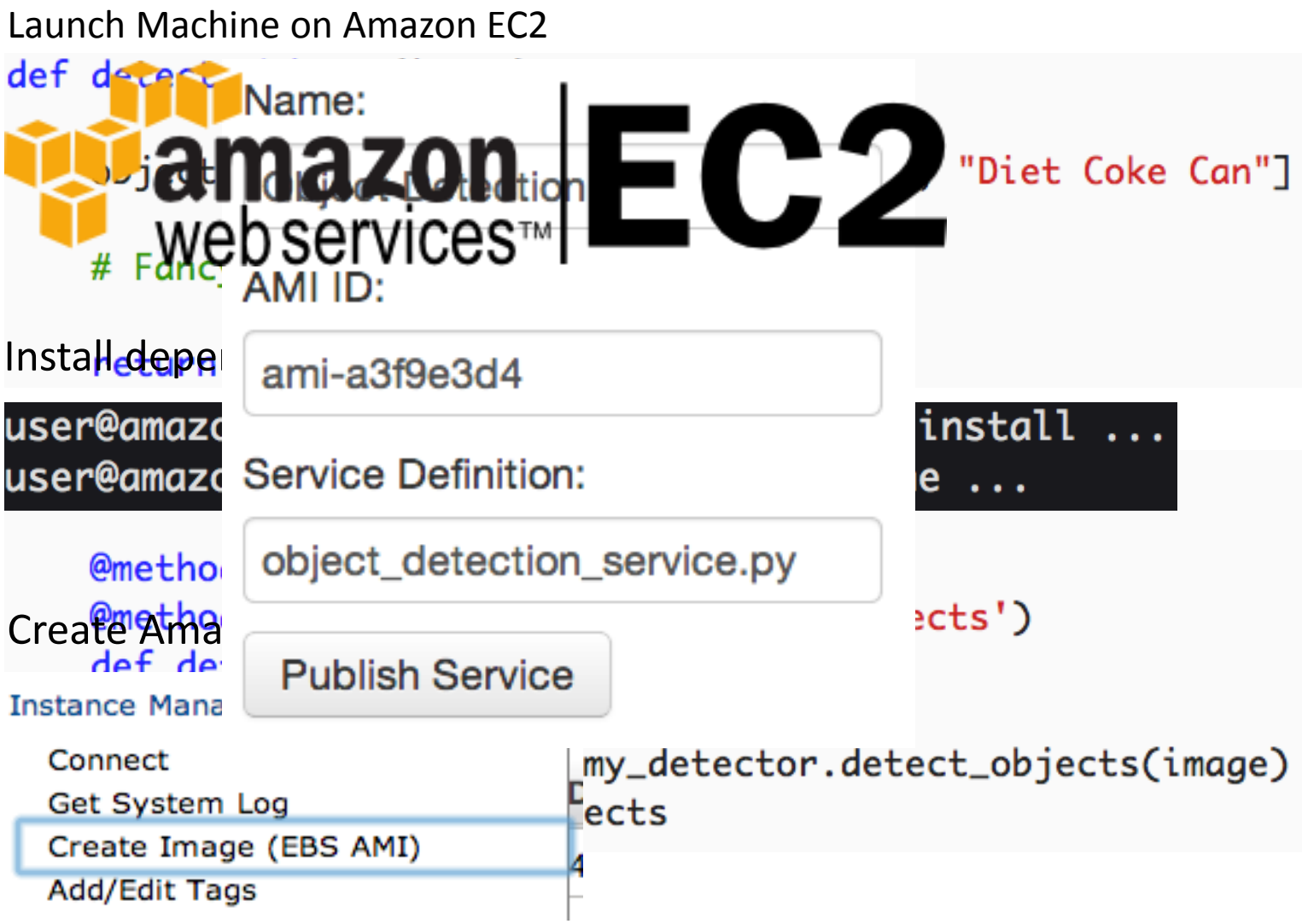
# RaaS Workflow

# RaaS Workflow – Algorithm Developers



# RaaS Workflow – Algorithm Developers

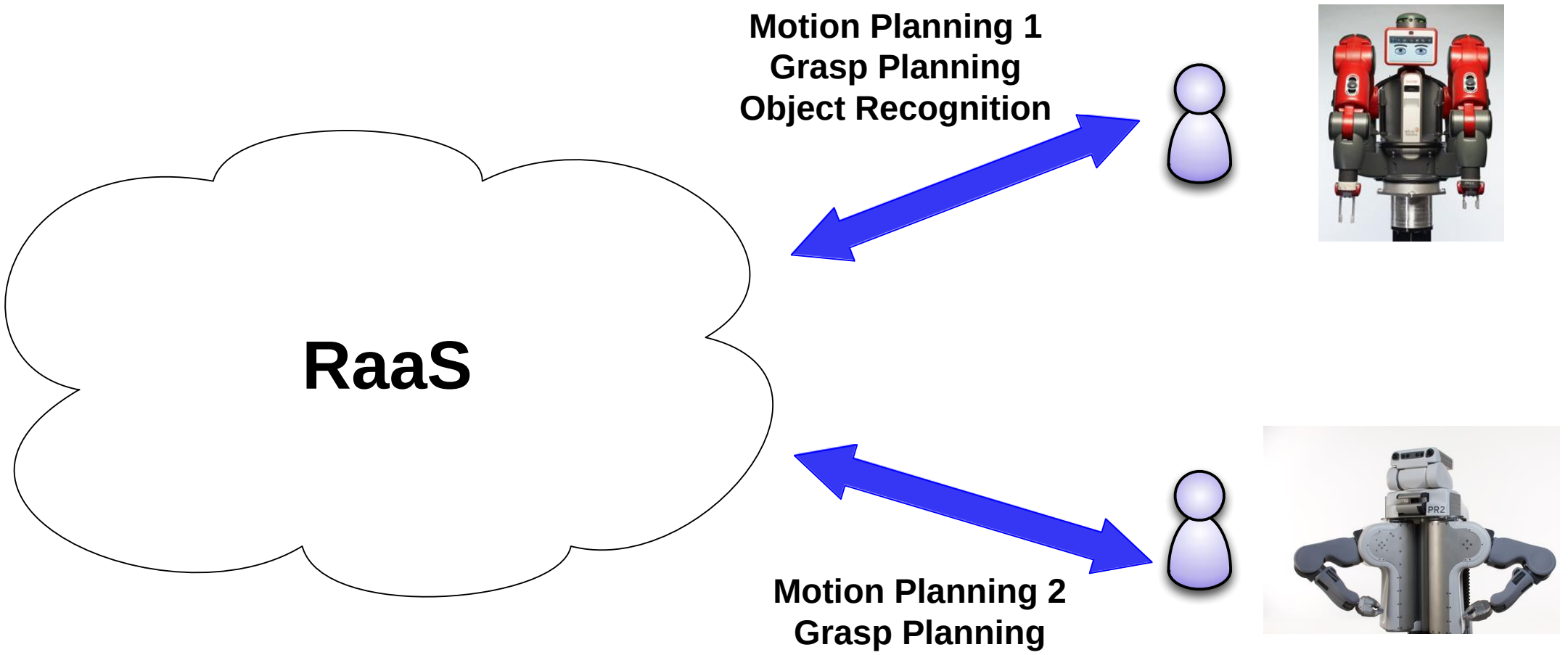
- Write your usual code
- Wrap with service code
- Create Amazon Machine Image (AMI)
- Publish your service



# RaaS Workflow – Algorithm Developers

- Make creating a web service as simple as possible
- Ignore HTTP, serialization, encoding, webservers, etc.
- Rich set of types available for service methods
  - Strings, floats, integers, binary blobs, raw files, timestamps, durations, poses, transformations, vectors, matrices, images, point clouds
  - Create your own
- Can serve ROS nodes as web services

# RaaS Workflow – Algorithm Users



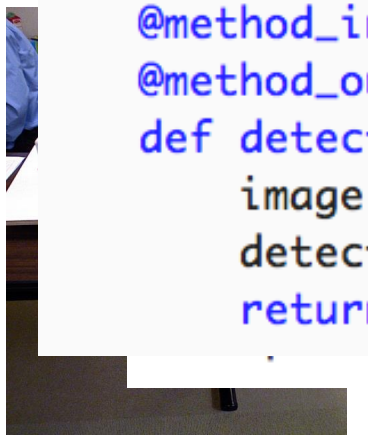
# RaaS Workflow – Algorithm Users

- Choose services
- Launch machines in the cloud
- Connect to machine and use services
- Detect objects in images, plan motions, etc.

**Service Definition:**

```
class ObjectDetectionService(Service):

    @method_input(Image, 'image')
    @method_output([String], 'detected_objects')
    def detect_objects(self,**inputs):
        image = inputs['image']
        detected_objects = my_detector.detect_objects(image)
        return detected_objects
```



Input Scene

**Client Code:**

```
service = connect('http://your-ec2-hostname/object_detection')
# de
dete
```

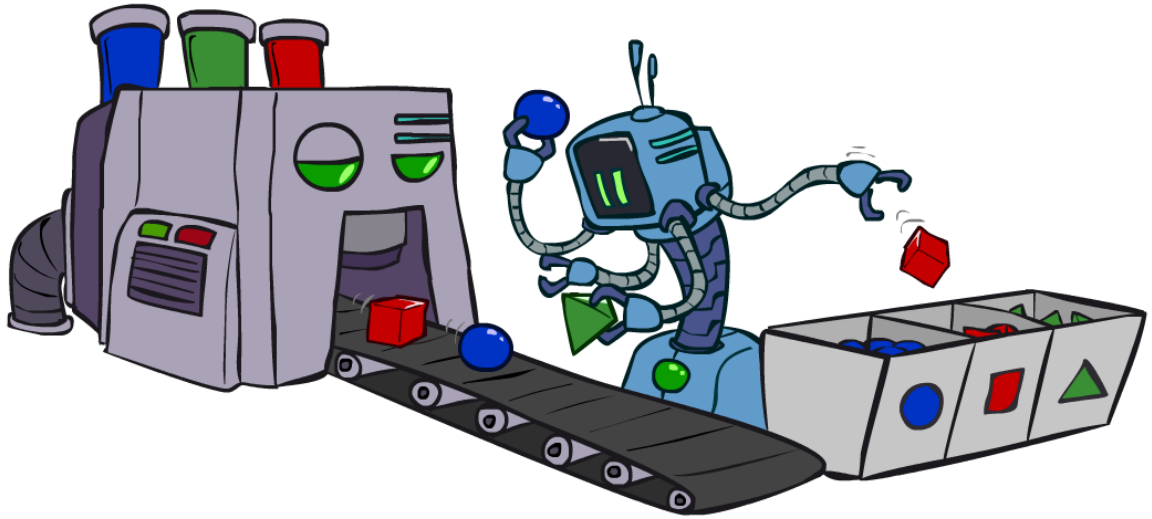

Detected Objects

# RaaS Workflow – Algorithm Users

- Compare different algorithms implementing the same interface by changing a single URL
- Again, ignore serialization, encoding/decoding, HTTP, etc. – use like an ordinary function call
- Use web-service-based ROS Nodes as if they were local ROS Nodes

# How to Solve It with RaaS

- No need to worry about computing hardware
- Install barebones ROS – (i.e. only messaging, low-level control, etc.)
- Use object recognition, motion planning, grasp planning services from RaaS
- RaaS complements ROS



Drawing: Ketrina Yim

# Collaborative Data Collection – Example

- Ten camera rig with controllable turntable
- High quality 3d models from about 600 images in 5 minutes of human time
- Ship us an object, we upload model to cloud, you can recognize it in images

# Roadmap

**(Almost) Complete (pre-alpha)**

- Service framework implemented for Python
- Service directory implemented for EC2
- Need to polish up rough edges and automate tedious steps

**Future Work**

- Linux containers rather than Amazon Machine Images
  - Can then run services anywhere – in your lab or in the cloud
- Infrastructure for other programming languages
- Data collection APIs

**Interested? Email us! arjun@eecs.berkeley.edu**