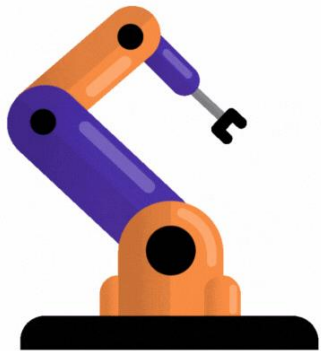




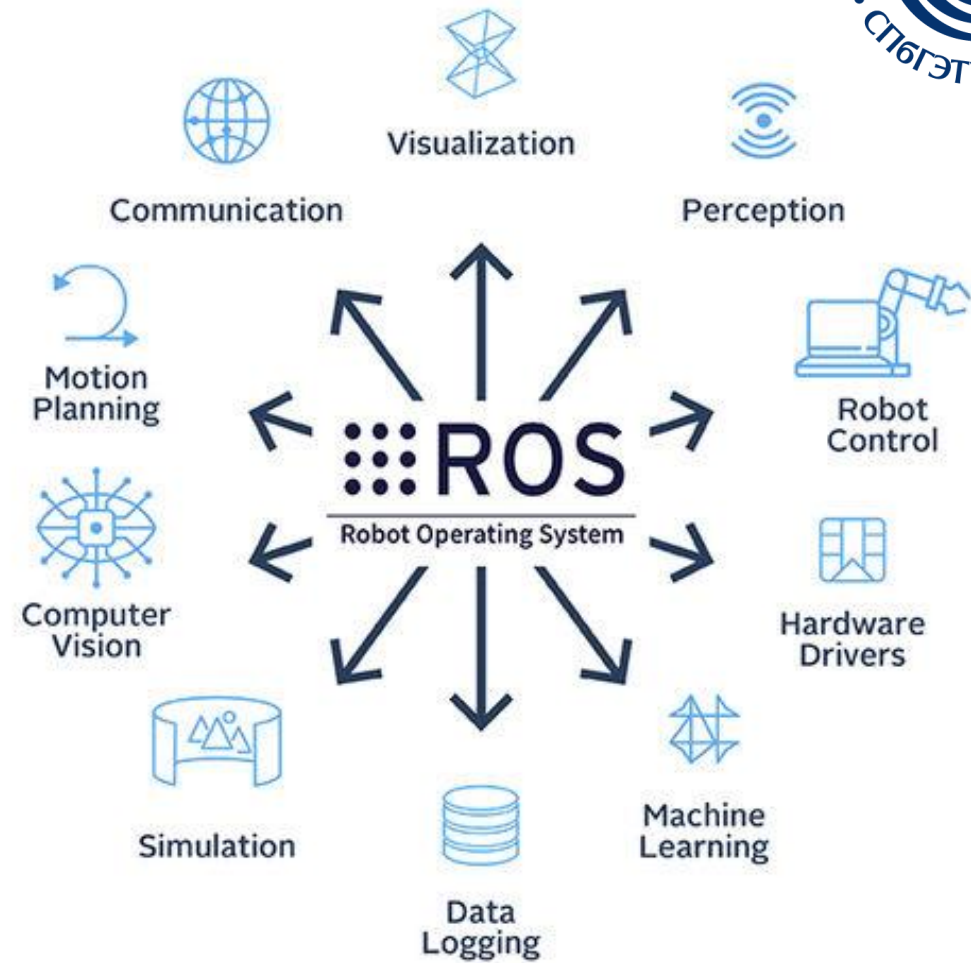
# Введение в ROS

Знакомимся с инструментом для работы  
с роботами

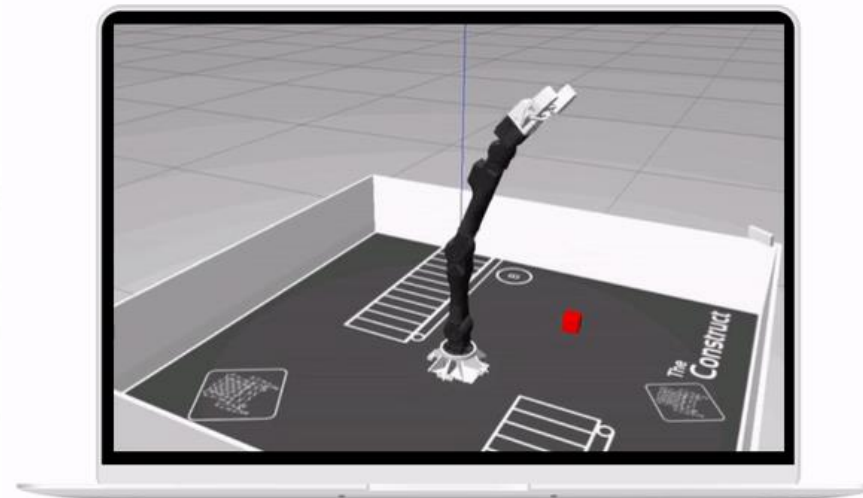
# Что такое ROS?



ROS

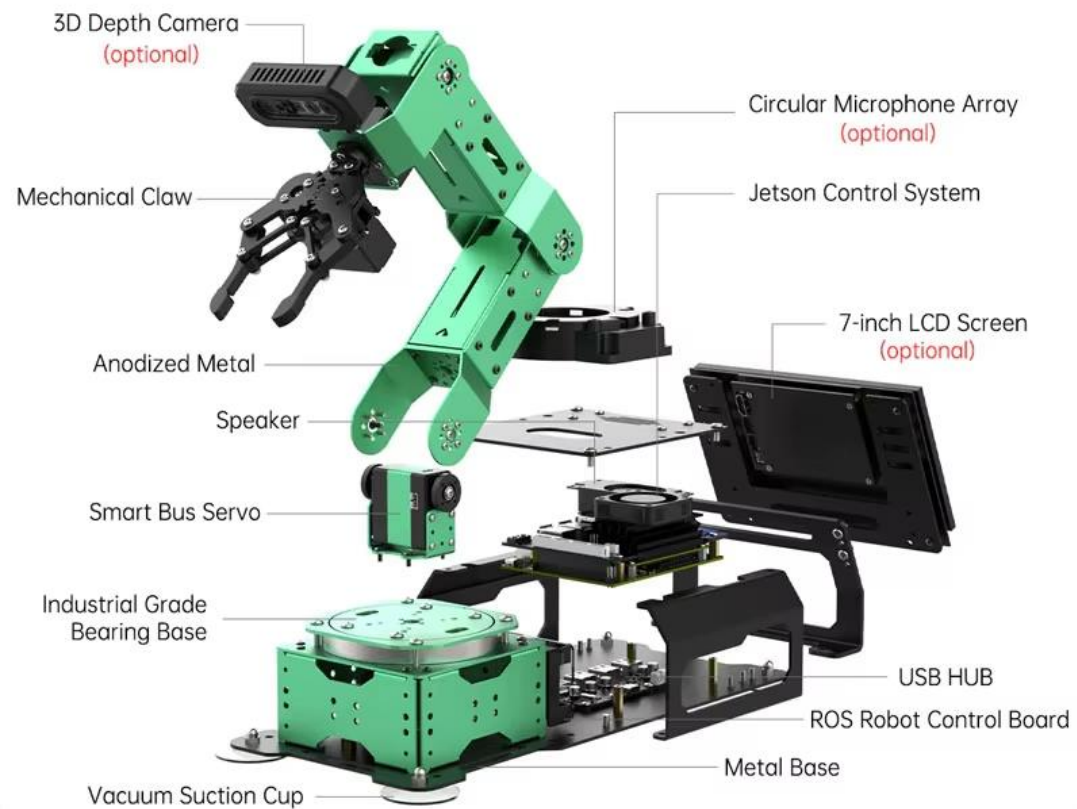


# Что такое ROS?



# Так зачем ROS?

## Product Structure





# Так зачем ROS?

Нагруженные задачи, но  
менее требовательны к  
стабильности



High Level -  
Software

ROS



Low Level -  
Software

RTOS



Простые задачи, но  
требовательны к  
стабильности

# Что такое ROS?



# ROS

Стандарт разработки

Инструменты разработки и отладки

Сообщество и OpenSource

# Стандарт коммуникации между программами



Датчик тока



Датчик скорости



Датчик  
температуры



?

Управление  
двигателем

# Стандарт коммуникации между программами



Датчик тока



Датчик скорости



Датчик  
температуры



Одна  
большая  
программа

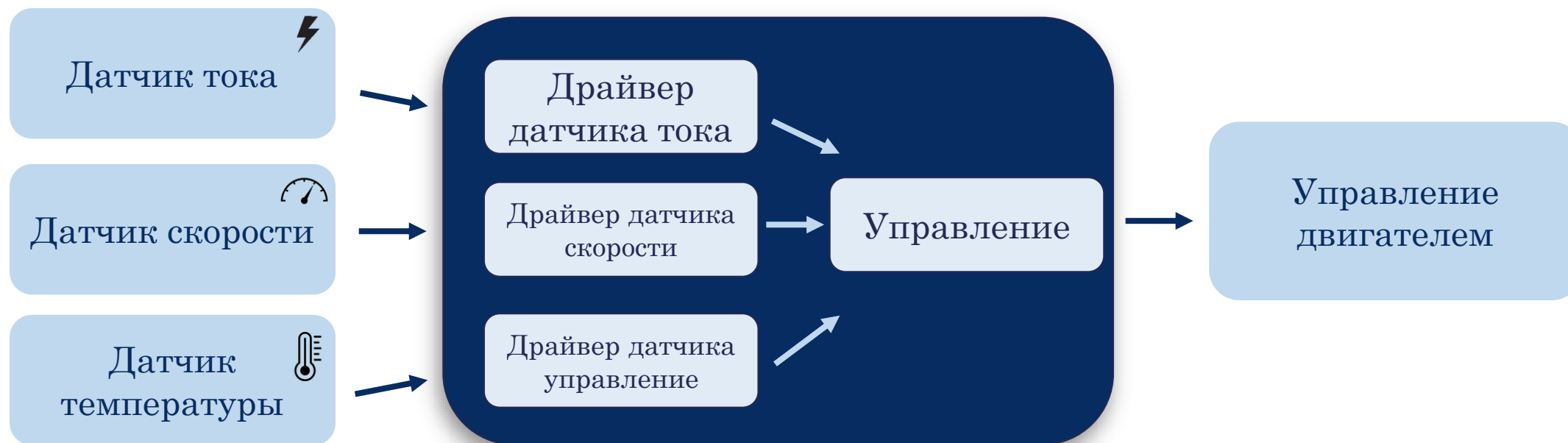
Управление  
двигателем



# Стандарт коммуникации между программами



Каждая связь  
описана своим  
форматом. Форматы  
могут быть общими



# Стандарт коммуникации между программами



Усложним – точки сбора информации и управления разделены



# Стандарт коммуникации между программами



Усложним – появляется больше датчиков



# Что такое ROS?



# ROS

Стандарт разработки

Инструменты разработки и  
отладки

Сообщество и OpenSource

# OpenSource – это круто!



- Возможность собрать под свою платформу
- Возможность просмотреть исходный код и понять логику работы
- Много готовых наработок - можно делать быстро базовые разработки, а более сложные кастомизировать не с нуля



# Основы узлов и топиков. Мастер

Мастер выполняет роль сервера имен для возможности подключения между собой различных нод. Команда **roscore** запускает сервер мастера, и после этого к нему могут подключиться и зарегистрироваться ноды ROS.

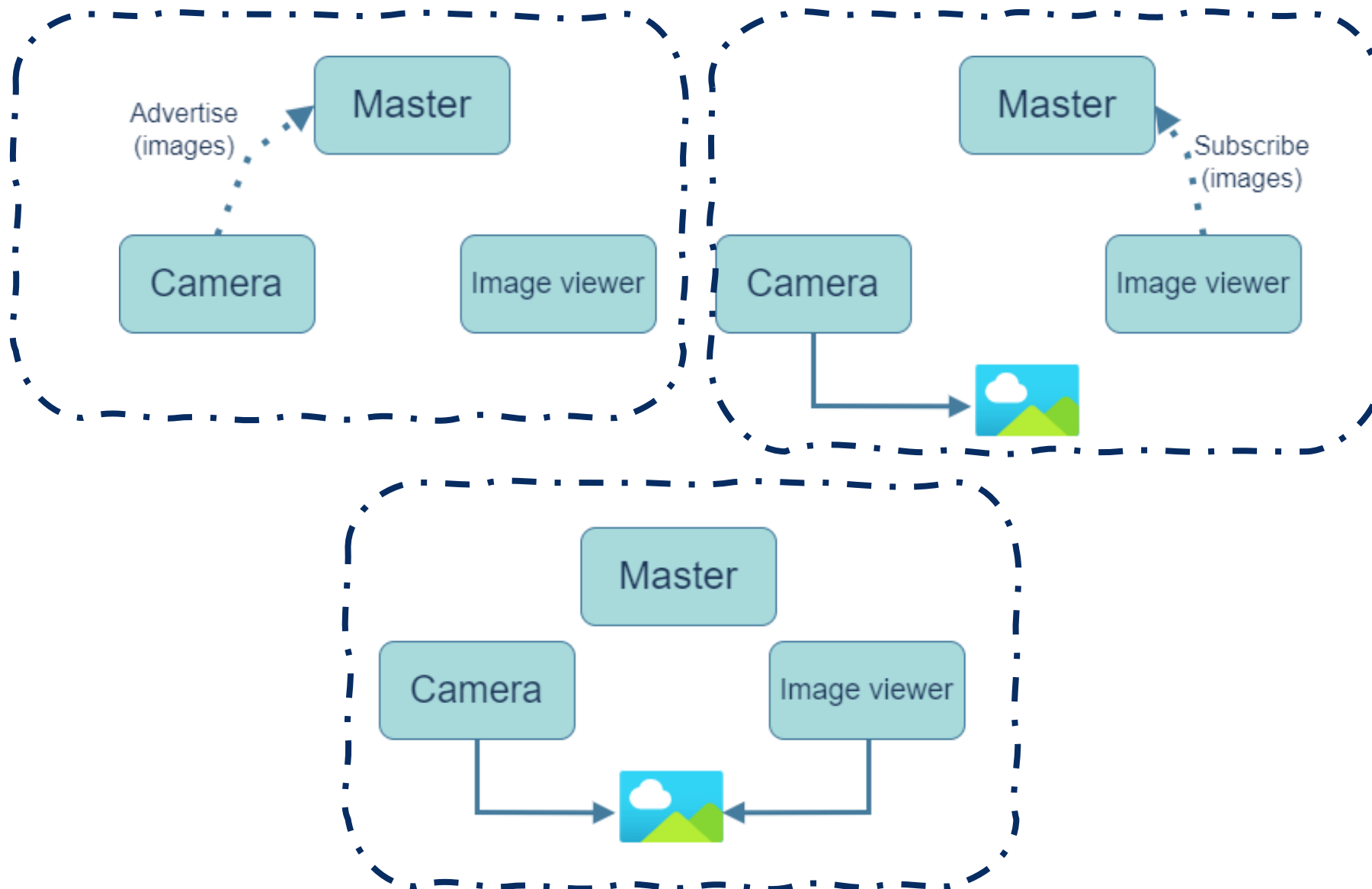


## Роль мастера:

- Позволить узлам определять местонахождения друг друга, используя протокол для подключения TCP/IP
- Отслеживание поступления в систему новых узлов

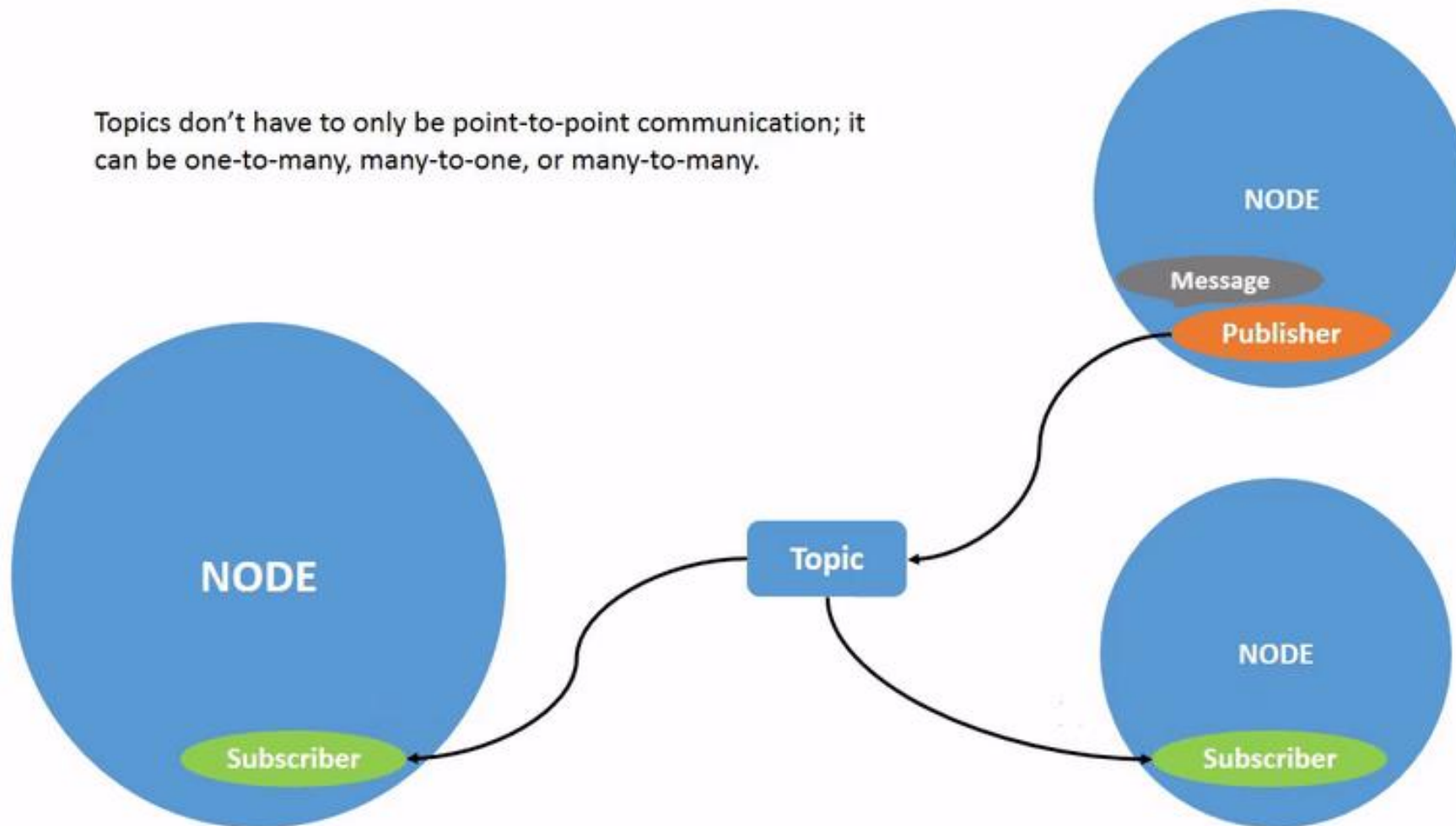


# Основы узлов и топиков. Мастер

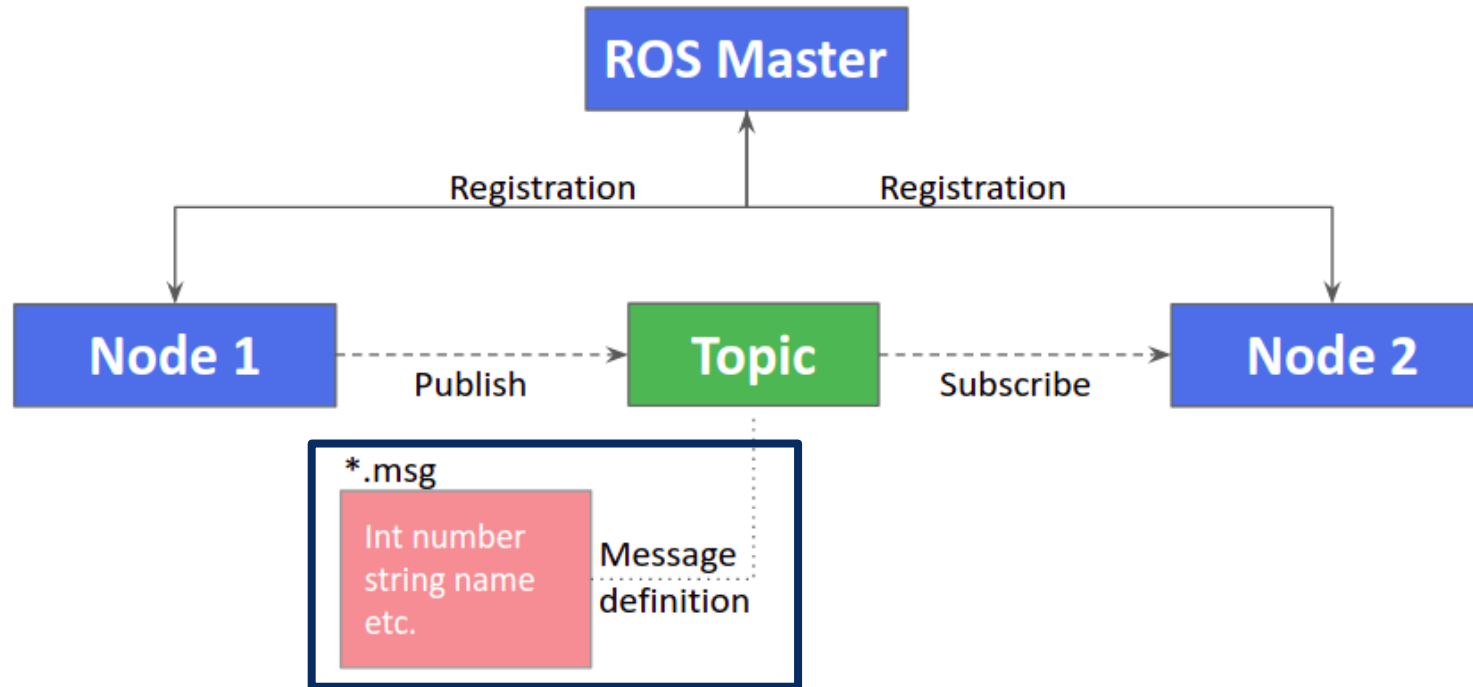


# Основы узлов и топиков

Topics don't have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.



# Messages



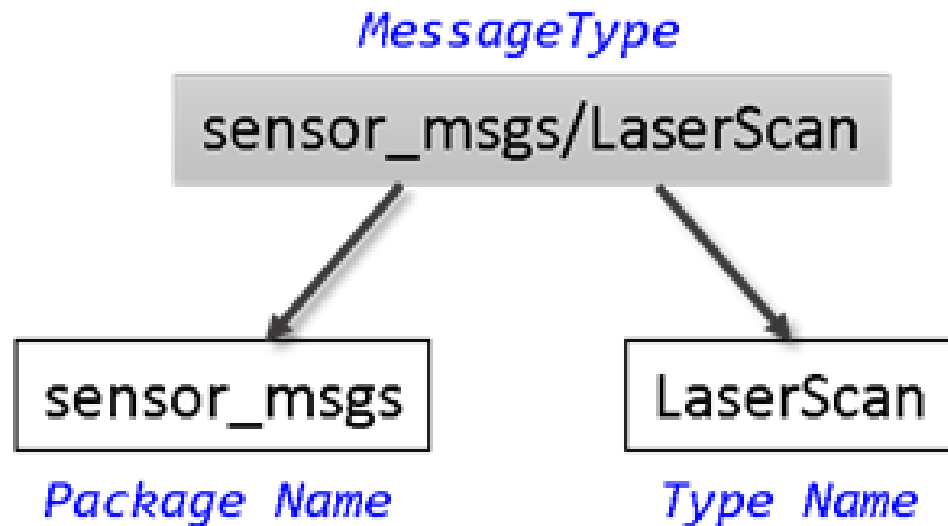
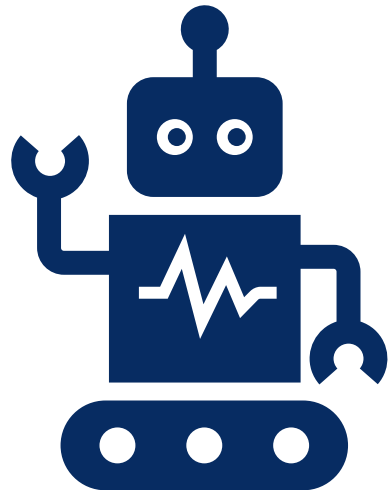
```
➔ ~ rostopic info /chatter
Type: std_msgs/String
Publishers:
* /talker (http://katya-pc:33435/)
Subscribers:
* /listener (http://katya-pc:45145/)
* /rostopic_9695_1660308932762 (http://katya-pc:40025/)
```

# Messages



Сообщение представляет собой структуру данных, которая используется при обмене информацией между узлами.

Топики (Topic), службы (Services) и Действия (Actions) используют сообщения для взаимодействия между собой. Сообщения могут включать в себя как базовые типы (целое число, число с плавающей точкой, логические и тд), так и массивы сообщений. Помимо этого, сообщения могут инкапсулировать в себе другие существующие типы сообщений и специальные заголовки.



# Launch файл



Launch - (eng) ввод в действие; ввод в эксплуатацию; пуск

```
vh@varhowto-com: ~  
tcl8.6-dev tk tk-dev tk8.6 tk8.6-bit2.5 tk8.6-dev  
ttf-bitstream-vera ttf-dejavu-core unixodbc-dev uuid-dev  
vtk7 wayland-protocols x11proto-core-dev x11proto-dev  
x11proto-input-dev x11proto-randr-dev  
x11proto-record-dev x11proto-scrnsaver-dev  
x11proto-xext-dev x11proto-xinerama-dev  
xorg-sgml-doctools xtrans-dev  
0 upgraded, 992 newly installed, 0 to remove and 49 not up  
graded.  
Need to get 529 MB of archives.  
After this operation, 2,706 MB of additional disk space will  
be used.  
Do you want to continue? [Y/n]
```

```
vh@varhowto-com: ~  
vh@varhowto-com:~$ sudo apt update  
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease  
Hit:2 http://dl.google.com/linux/chrome/deb stable InRelease  
Hit:3 http://packages.ros.org/ros/ubuntu focal InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease  
Hit:5 https://download.docker.com/linux/ubuntu bionic InRelease  
Hit:6 http://archive.ubuntu.com/ubuntu focal-updates InRelease  
Hit:7 http://archive.ubuntu.com/ubuntu focal-backports InRelease  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
48 packages can be upgraded. Run 'apt list --upgradable' to see them.  
vh@varhowto-com:~$  
vh@varhowto-com:~$
```

TERMINALS HELL

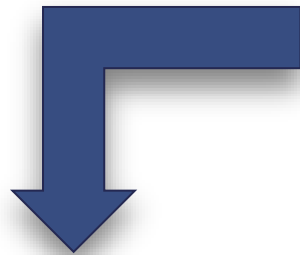
```
Terminal  
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml  
Skip end-of-life distro "ardent"  
Skip end-of-life distro "bouncy"  
Skip end-of-life distro "crystal"  
Skip end-of-life distro "dashing"  
Skip end-of-life distro "eloquent"  
Skip end-of-life distro "foxy"  
Skip end-of-life distro "galactic"  
Skip end-of-life distro "groovy"  
Add distro "humble"  
Skip end-of-life distro "hydro"  
Skip end-of-life distro "indigo"  
Add distro "iron"  
Skip end-of-life distro "jade"  
Skip end-of-life distro "kinetic"  
Skip end-of-life distro "lunar"  
Skip end-of-life distro "melodic"  
Add distro "noetic"  
Add distro "rolling"  
updated cache in /home/admin/.ros/rosdep/sources.cache  
* Stopping hotplug events dispatcher systemd-udevd [ OK ]  
* Starting hotplug events dispatcher systemd-udevd [ OK ]  
admin@mandalore22:~/workspaces/isaac_ros-dev$
```

```
vh@varhowto-com: ~  
vh@varhowto-com:~$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E68ADE8868B172B4F42ED6FBAB17C654  
Executing: /tmp/apt-key-gpghome.mdMVphTCAR/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv-key C1CF6E31E68ADE8868B172B4F42ED6FBAB17C654  
gpg: key F42ED6FBAB17C654: public key "Open Robotics <info@osrfoundation.org>" imported  
gpg: Total number processed: 1  
gpg: imported: 1  
vh@varhowto-com:~$
```

```
roscore http://varhowto-com:11311/  
vh@varhowto-com:~/opt/ros/noetic$ roscore  
... logging to /home/vh/.ros/log/b30d8df6-9eee-11ea-906d-492a6c9dca07/roslaunch-varhowto-com-26839.log  
Checking log directory for disk usage. This may take a while  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://varhowto-com:37317/  
ros_comm version 1.15.6  
  
SUMMARY  
=====  
PARAMETERS  
* /roscore: noetic  
* /rosversion: 1.15.6  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [26864]  
ROS_MASTER_URI=http://varhowto-com:11311/  
  
setting /run_id to b30d8df6-9eee-11ea-906d-492a6c9dca07  
process[roscout-1]: started with pid [26874]  
started core service [/roscout]
```

# Launch система

```
roscore
roslaunch ...
roslaunch ...
roslaunch ...
...
```



launch-файл

```
<launch>

  <node>
    ...
  </node>

  <node ... />

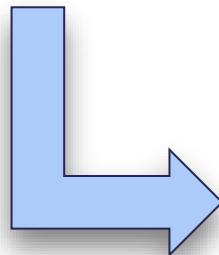
  <node>
    ...
  </node>
</launch>
```

roslaunch ... - все в одном терминале



# Launch-file пример

```
<launch>  
  <node name="listener" pkg="rospy_tutorials" type="listener.py" output="screen"/>  
  <node name="talker" pkg="rospy_tutorials" type="talker.py" output="screen"/>  
</launch>
```



```
roslaunch study_pkg  
sample_launch.launch
```

```
PARAMETERS  
* /roscdistro: noetic  
* /rosversion: 1.15.14  
  
NODES  
/  
  listener (rospy_tutorials/listener.py)  
  talker (rospy_tutorials/talker.py)  
  
auto-starting new master  
process[master]: started with pid [1025]  
ROS_MASTER_URI=http://localhost:11311  
  
setting /run_id to e5b7943c-bb73-11ed-9ddb-00155db038a2  
process[rosout-1]: started with pid [1054]  
started core service [/rosout]  
process[listener-2]: started with pid [1057]  
process[talker-3]: started with pid [1058]
```

# Launch vs scripts

- XML формат
- Не регламентирует порядок =(
- Просто описывает список того, что должно быть запущено и как должно быть запущено
- Используется для структурирования и описания запуска
- Содержат очень мало логических конструкций (например, нет циклов)

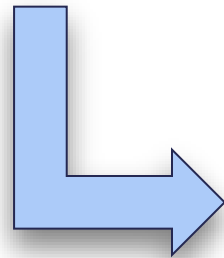
# Launch и мапинг топиков



```
<launch>
  <node name="listener" pkg="rospy_tutorials" type="listener.py">
    <remap from="chatter" to="chat_channel" />
  </node>
  <node name="talker" pkg="rospy_tutorials" type="talker.py">
    <remap from="chatter" to="chat_channel" />
  </node>
</launch>
```

# Launch и пространства имен

```
<launch>
  <group ns="my_ns">
    <node name="listener" pkg="rospy_tutorials" type="listener.py" />
    <node name="talker" pkg="rospy_tutorials" type="talker.py" />
  </group>
</launch>
```



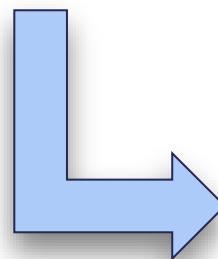
```
$ rosnode list
/my_ns/listener
/my_ns/talker
/rosout
```

```
$ rostopic list
/my_ns/chatter
/rosout
/rosout_agg
```

# Launch аргументы

```
<launch>
  <arg name="nodes_suffix" default="" />

  <node name="listener_$(arg nodes_suffix)" pkg="rospy_tutorials" type="listener.py" />
  <node name="talker_$(arg nodes_suffix)" pkg="rospy_tutorials" type="talker.py" />
</launch>
```



```
$ rosnode
list
/listener_123
/rosout
/talker_123
```

```
roslaunch study_pkg sample_launch_args.launch nodes_suffix:=123
```

# Launch in da launch

```
<launch>
  <include file="$(find study_pkg)/launch/start_talker.launch"/>
  <include file="$(find study_pkg)/launch/start_listener.launch"/>
</launch>
```

launch/start\_listener.launch

```
<launch>
  <node name="listener" pkg="rospy_tutorials" type="listener.py">
  </node>
</launch>
```

launch/start\_talker.launch

```
<launch>
  <node name="talker" pkg="rospy_tutorials" type="talker.py">
  </node>
</launch>
```



# Заметочки

- Разделяйте большой launch файл на малые файлы и переиспользуйте через включения
- Для файлов, которые не должны быть напрямую вызваны, используйте префикс “\_” (например, \_start\_gmapping.launch)
- В имена файлов полностью вписывайте контекст (директории при запуске не учитываются – `roslaunch pkg <имя файла без пути>`), например, `gazebo_start_slam.launch`