



# Roscon + iROS 2019 Macau



# About Mov.ai

- Helping robot developers
  - Quick goto market + long-term support
  - Visual launch of ROS Node network, in-browser IDE, SW Distribution, reversible upgrades, multi-protocol event processor
- Helping industrial operators and automation integrators
  - Tools to set up fleets of robots of different types

## **Commercial Grade ROS** | ROS for Business

- VC funded startup, since 2016
- Team developed custom autonomous mobile robots for 7 years
- 20 engineers, based in Lisbon
- 2 AGV partners - cart moving (TUGBOT) and pallet moving (RPM)
- Launching in Macau ROSCON 2019

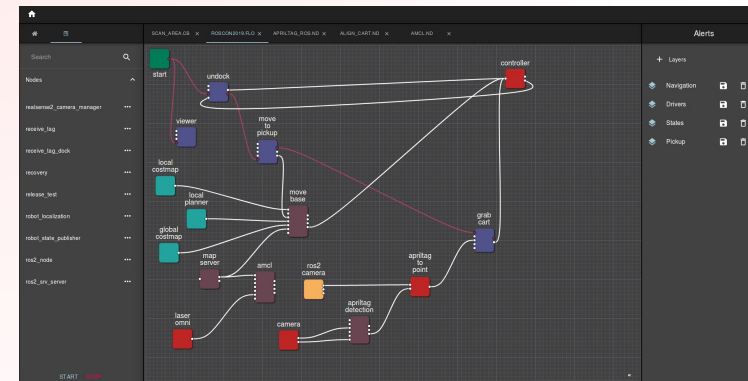
## **Inviting ROS community to join MOV.AI the Beta program**

# Visual Launch System for ROS

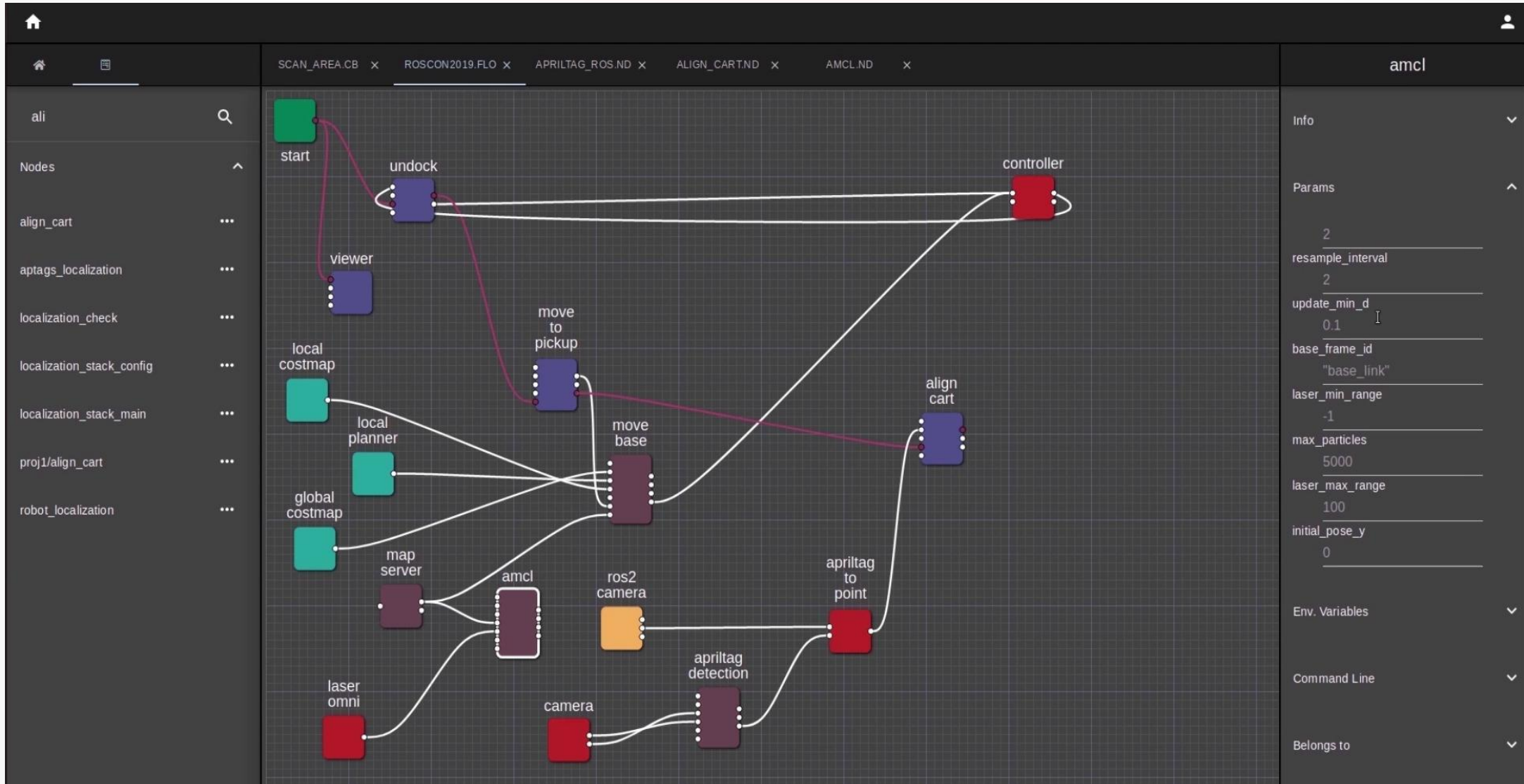
- Replaces roslaunch / rosruntime framework with **Visual Launch Diagrams**
- Drag & Drop Nodes, Connect Node inputs/outputs
- **VLD** lines represent communication protocols between nodes
  - Modify Nodes/connections in split seconds
  - Organize multiple Node networks, Node versions & Parameters
  - Supports all ROS protocols, TF, Nodelets, pluginlib

```

1 <launch>
2 <!-- Turtlesim Node-->
3 <node pkg="turtlesim" type="turtlesim_node" name="sim"/>
4 <node pkg="turtlesim" type="turtle_teleop_key" name="teleop"
  output="screen"/>
5 <node pkg="learning_tf" type="turtle_tf_listener.py"
6   name="listener" />
7
8 <node name="turtle1_tf_broadcaster" pkg="learning_tf"
9   type="turtle_tf_broadcaster.py" respawn="false" output="screen" >
10 <param name="turtle" type="string" value="turtle1" />
11 </node>
12 <node name="turtle2_tf_broadcaster" pkg="learning_tf"
13   type="turtle_tf_broadcaster.py" respawn="false" output="screen" >
14 <param name="turtle" type="string" value="turtle2" />
15 </node>
16 </launch>
    
```



# Visual Launch System for ROS



# Node Setup for Visual Launch



- Data required to deploy existing ROS node in Visual Launcher
- Input / Output ports (pub, sub, action, service)
- Parameters - command line, parameter server, environment vars
- Default parameters (template), drag into a Flow (instance)
- Support for multiple versions of same Node
- Migration tools - to help in input all the Node related data

# Node Setup for Visual Launch



The screenshot displays the ROS Visual Launch Editor interface. At the top, several workspace tabs are visible: SCAN\_AREA.CB, ROSCON2019.FLO, APRILTAG\_ROS.ND, ALIGN\_CART.ND, and AMCL.ND. The main workspace is divided into three sections: Information, Parameters, and I/O Configuration.

**Information Section:**

- Name: `amcl_test`
- Persistent ⓘ
- Dummy ⓘ
- Description: AMCL is a probabilistic localization system for a robot moving in 2D. It implements the adaptive (or KLD-sampling) Monte Carlo localization approach which uses a particle filter to track the pose of a robot against a known map.

**I/O Configuration Section:**

Name	Transport / Protocol	Package	Message	Actions
> particlecloud	ROS1/Publisher	geometry_msgs	PoseArray	
> static_map	ROS1/ServiceClient	nav_msgs	GetMap	
> set_map	ROS1/ServiceServer	nav_msgs	SetMap	



# MOV.AI Nodes - IDE in Browser



- Multi-Protocol Event processor
  - Message/Event triggers Callback in Python
  - Support for ROS1, ROS2, HTTP, WebSocket, Serial Driver, Redis DB..
  - Callback code cannot access communication layer
- Native parallel processing
  - Callbacks are Re-entrant - Persistent data only via Redis DB API
  - AsyncIO backend + Cython (C level performance)
  - Resource Usage Profiling tools
- Upgrade / Downgrade - mandatory for industrial clients
  - Imported libraries - outside the callback code
  - GIT based versioning of Callbacks

# MOV.AI Nodes - IDE in Browser



The screenshot shows the MOV.AI IDE interface. On the left is a sidebar with a search bar and a list of project components: Annotation, Callbacks, Flows, State Machines, Nodes, Layouts, Scenes, and Forms. The main area is a code editor with a dark background, showing Python code for a ROS node named 'scan\_area'. The code includes comments and logic for processing laser scan data. A context menu is open over the code, listing options: 'set' (local), 'share' (local), 'save' (local), 'database' (local), and 'samples' (local). The right sidebar shows the 'scan\_area' node selected, with sections for 'Info', 'Imports', and 'Message'.

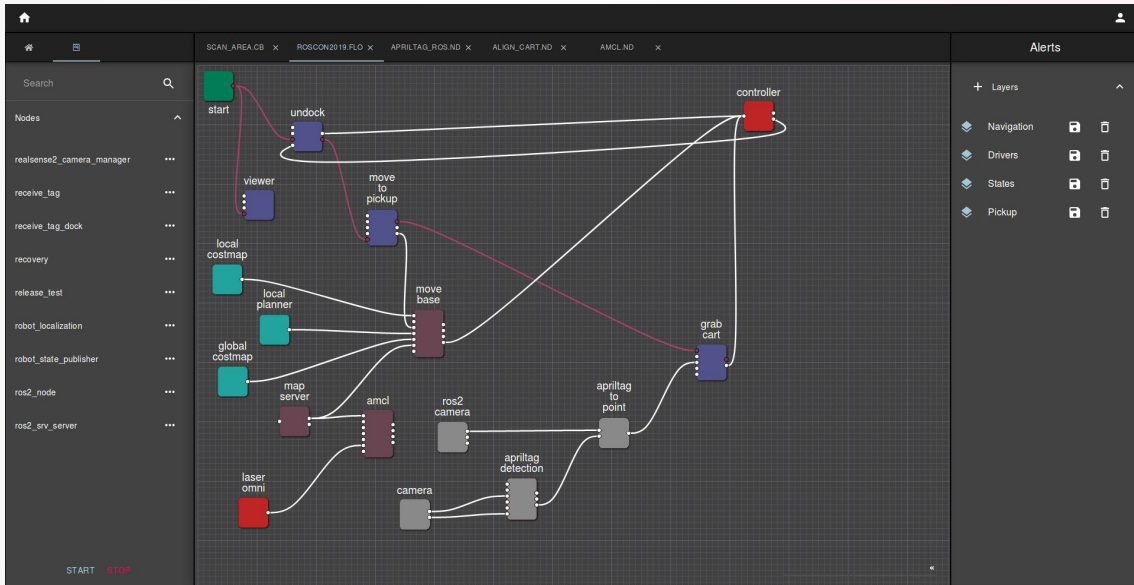


# State Transition

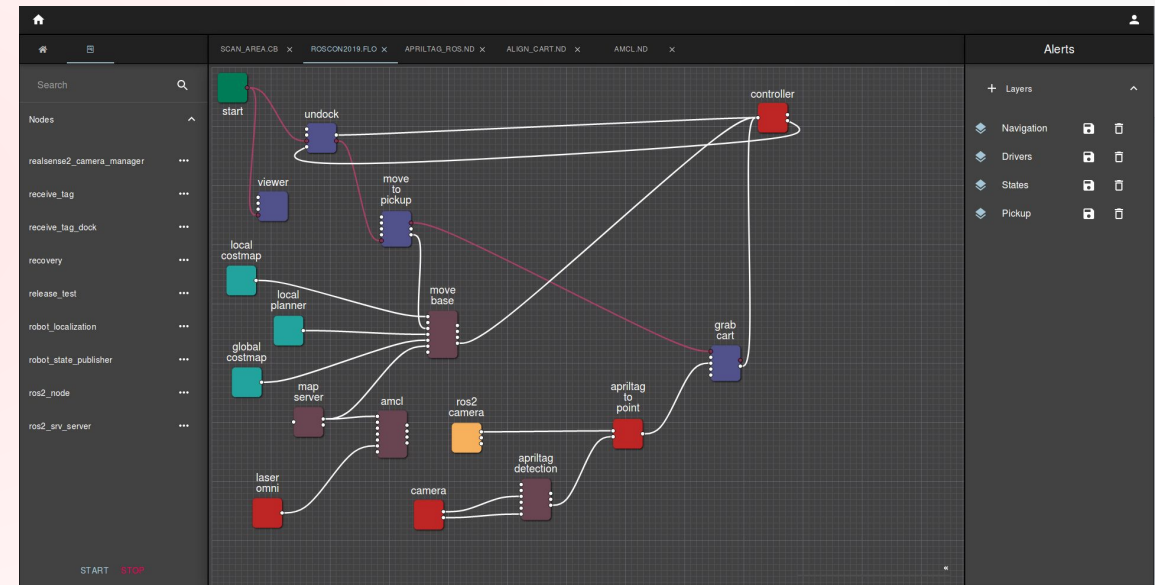


- Visual Launch Diagrams can act as State Machine Diagrams
- MOV.AI Nodes can act as “State Nodes”
  - “Transition” is one of the supported VLD protocols
- When a Node is transitioned-to
  - All non-connected nodes are recursively killed/disabled
- Visualize dependency between Robot’s State & required ROS nodes
- Visual ROS2 lifecycle manager

# State Transition



Corridor navigation state



Cart Grabbing state

# Customizable UI

- MVC framework
  - Collaborative UI (2-direct. link w db replicated on every robot)
  - Any Robot can act as Web server
- Extensible web API
  - HTTP/WebSockets protocols supported in MOV.AI Nodes
  - Custom server-side functions
  - REST API available for CRUD actions (vs replicated DB)
- REST-full application
  - Develop apps with preferred js framework (React, Vue, etc)
  - Create and upload your own js application into Mov.AI system
- Dashboard Creator
  - Create operator views with stats and queues
  - Create custom dashboards to monitor and control fleets
  - Extensible set of customizable widgets

# Dashboard Creator



UNTITLED.LY x

Demonstration

### Joystick

robotA 20Hz

Connected

### StartFlow

START STOP

### Video

### AGVStatus

Tugbot

Battery: 100%

GO WORK

Current Task: Carrying Palette

Date	Time	Description
------	------	-------------

### Video Props

x: 4

y: 0

i: 3

Width: 5

Height: 17

minWidth: 3

minHeight: 8

URL: http://10.10.0.233:8080/stre...

Video Type: Stream

SAVE LAYOUT

SAVE

Developer

EXPORT APP

**Thank you!**

The logo for MOV.ai is centered on a white circular background. The word "MOV" is in a stylized, bold, red font, and ".ai" is in a smaller, red, sans-serif font. The background features a faint, circular pattern of binary code (0s and 1s) and a network of grey lines with dots, resembling a circuit board or data flow diagram.

**MOV.ai**

[www.mov.ai](http://www.mov.ai)  
[info@mov.ai](mailto:info@mov.ai)