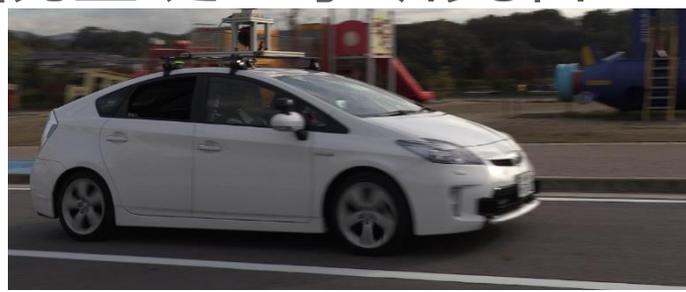




# 自動運转向けソフトウェアAutowareと MATLAB<sup>®</sup>/Simulink<sup>®</sup>の連携

安積卓也

大阪大学大学院基礎工学研究科



# 自動運転のレベル

※ 1 官民ITS構想RM用語対応表  
操舵：ハンドル（ステアリング）  
加速：アクセル  
制動：ブレーキ

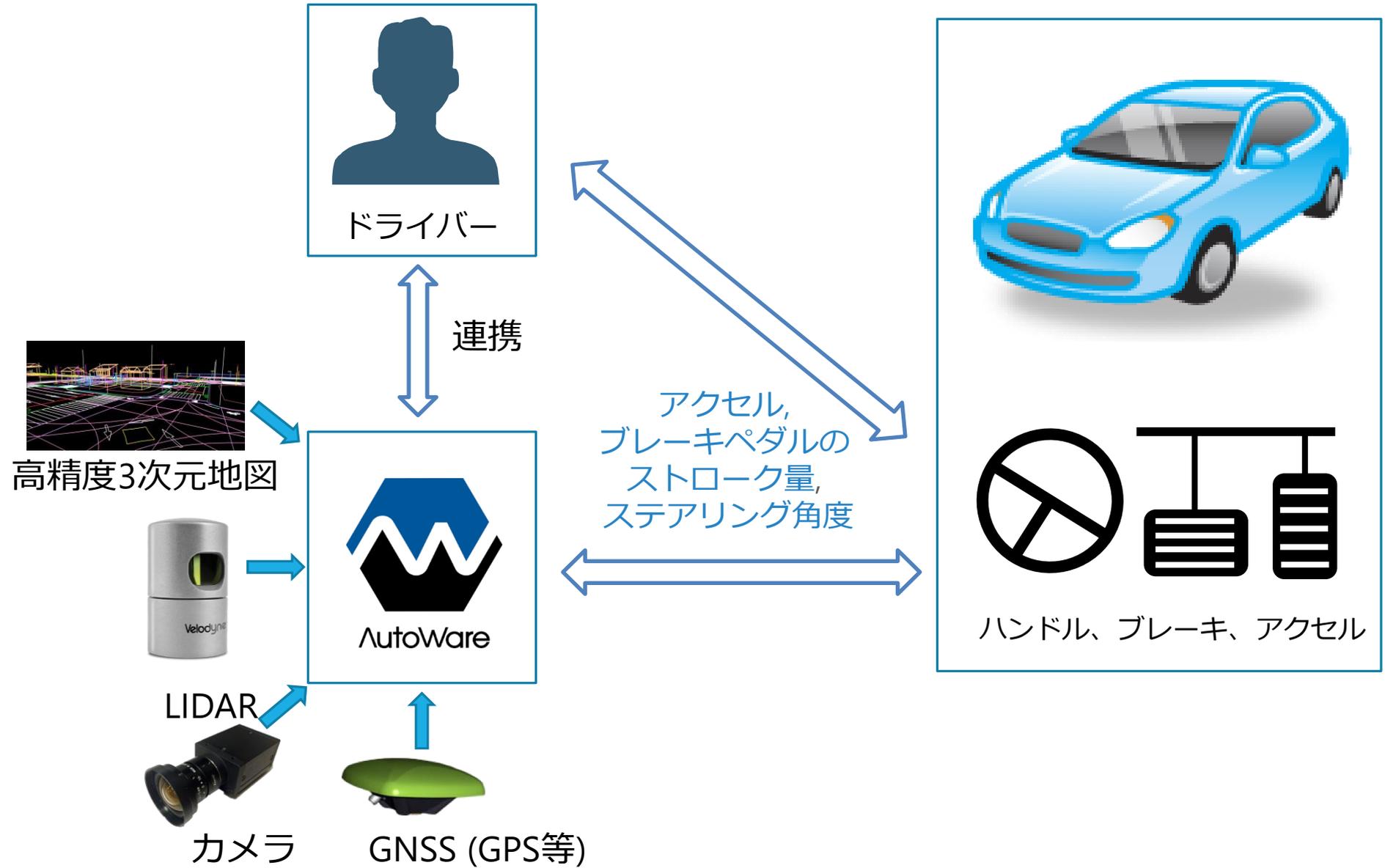
レベル	システム：※1 ハンドル・アクセル・ブレーキ	ドライバー	場所	
レベル1	いずれか一つ	主体		自動ブレーキ
レベル2	複数	主体		クルーズ コントロール
レベル3	すべて (危険回避はドライバー)	あり		公道実験
レベル4	すべて	なし	限定	自動運転バス
レベル5	すべて	なし	全て	自動運転 タクシー

完全自動運転

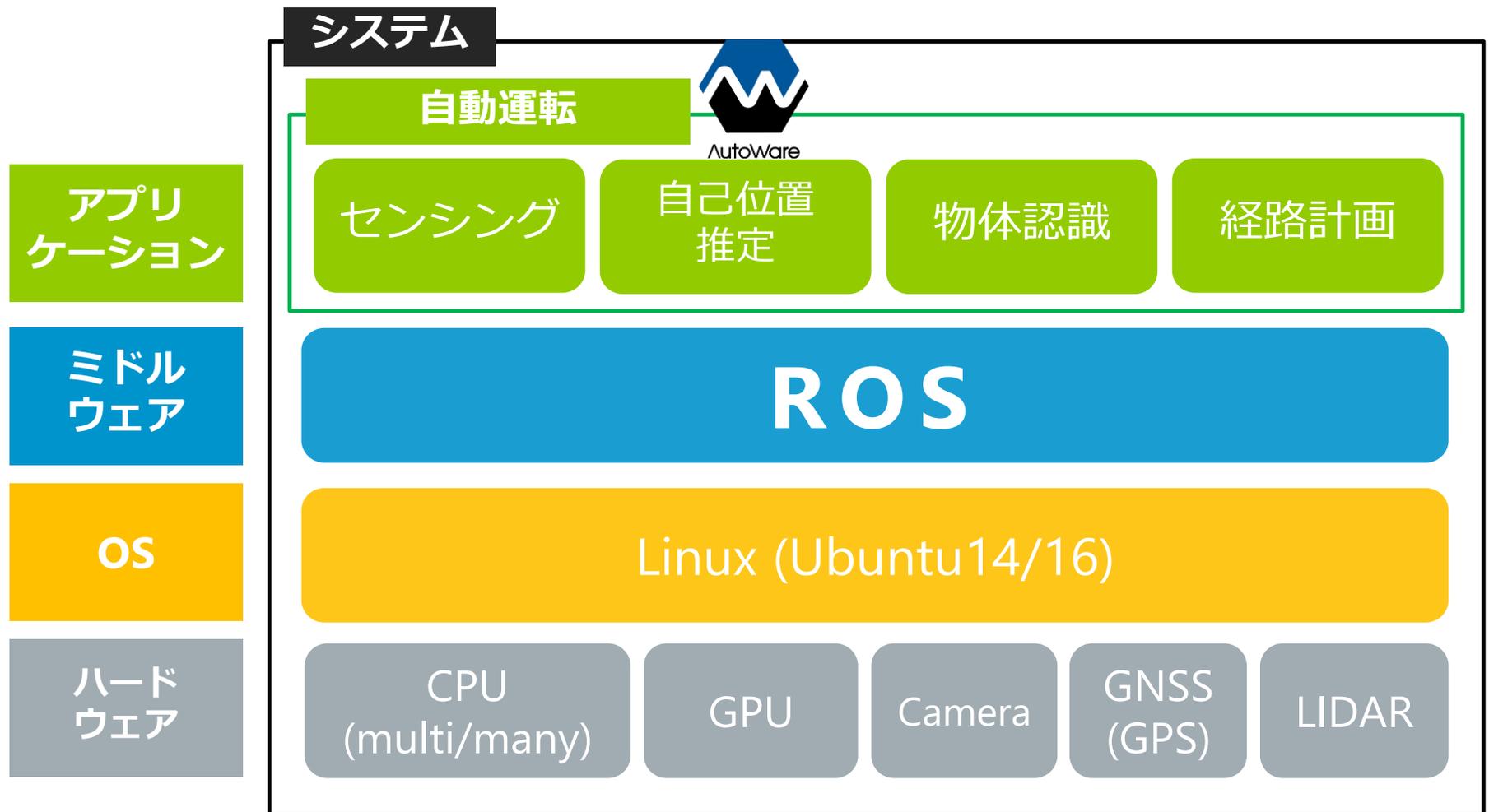
※SAE J3016 (2016)



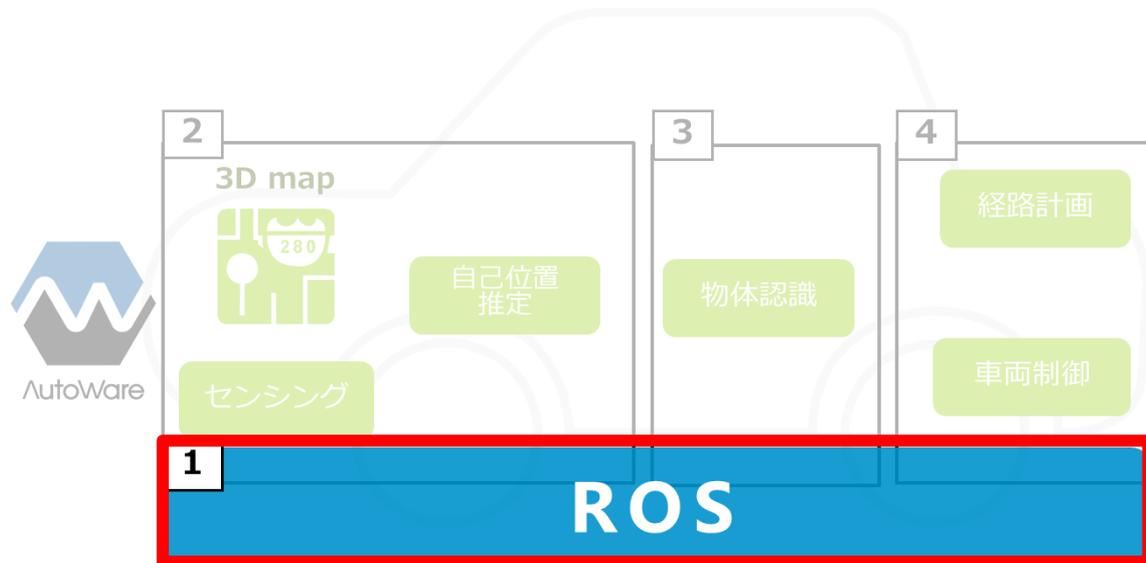
# 自動運転ソフトウェアの役目



# Autowareの構成



<https://github.com/CPFL/Autoware>



ROS (Robot Operating System)

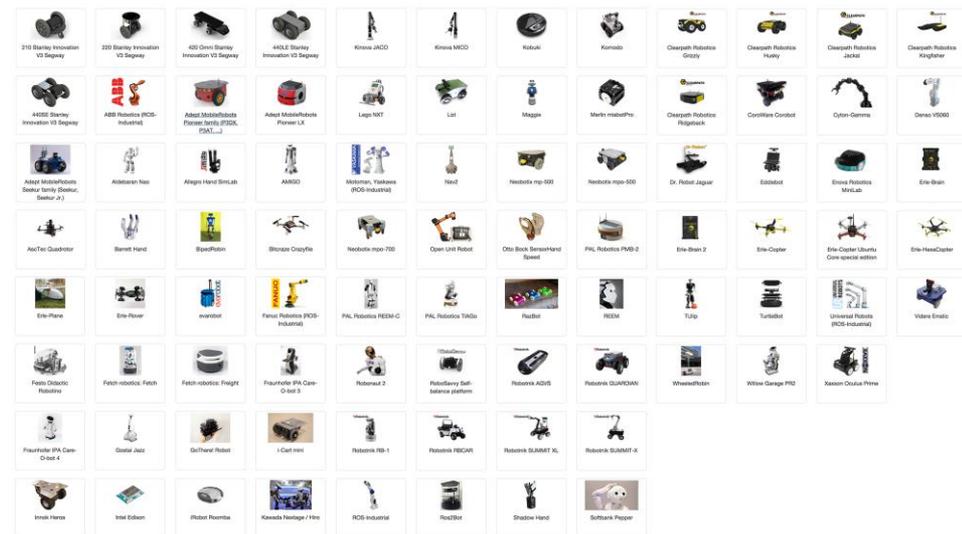
## ROS (Robot Operating System)

: ロボット開発におけるライブラリやツールを提供

ハードウェアの抽象化、デバイスドライバ、ライブラリ、視覚化ツール、データ通信、パッケージ管理 ...etc

## 特長

- 世界で最も利用されているロボットミドルウェア
- 豊富な対応ロボット・センサ
- オープンソース
- サポート言語 : C++, Python
- 管理団体 : OSRF
- 対応OS : Linux



# ROS の 特長

## ROS (Robot Operating System)

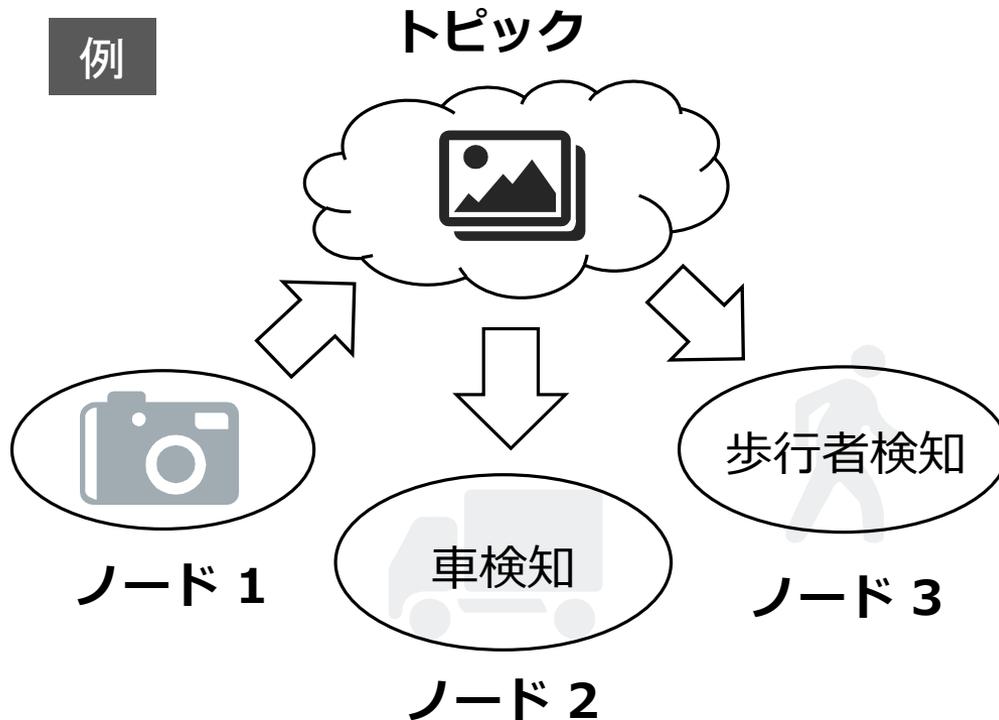
: ロボット開発におけるライブラリやツールを提供



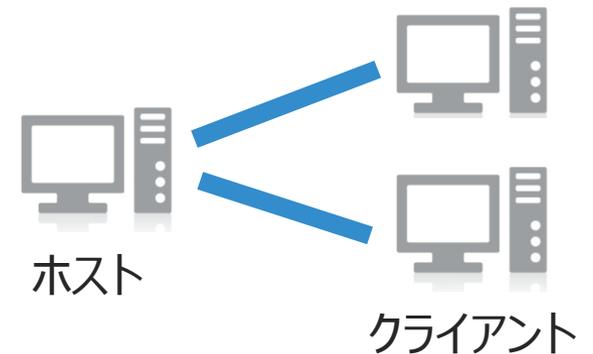
### Publish / Subscribe モデル

- ノードの集合としてシステムを構築
- トピックを介してデータをやり取り

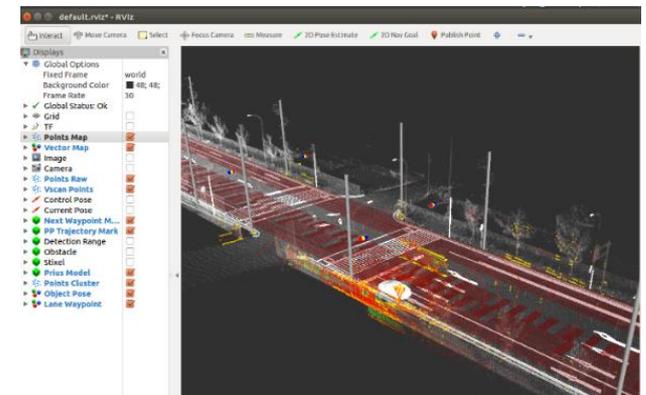
例



### 分散システム



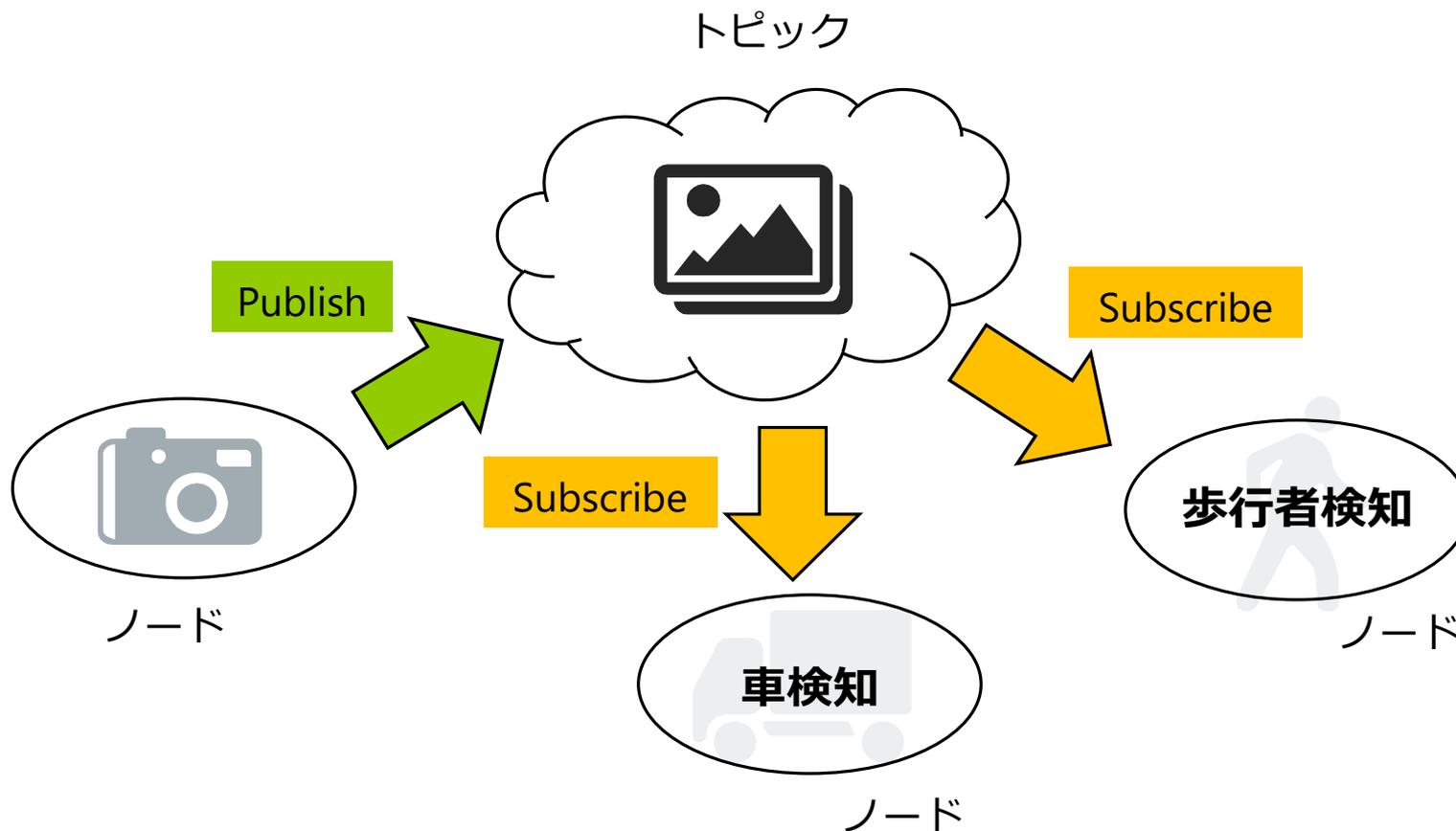
### 視覚化・シミュレーション



# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

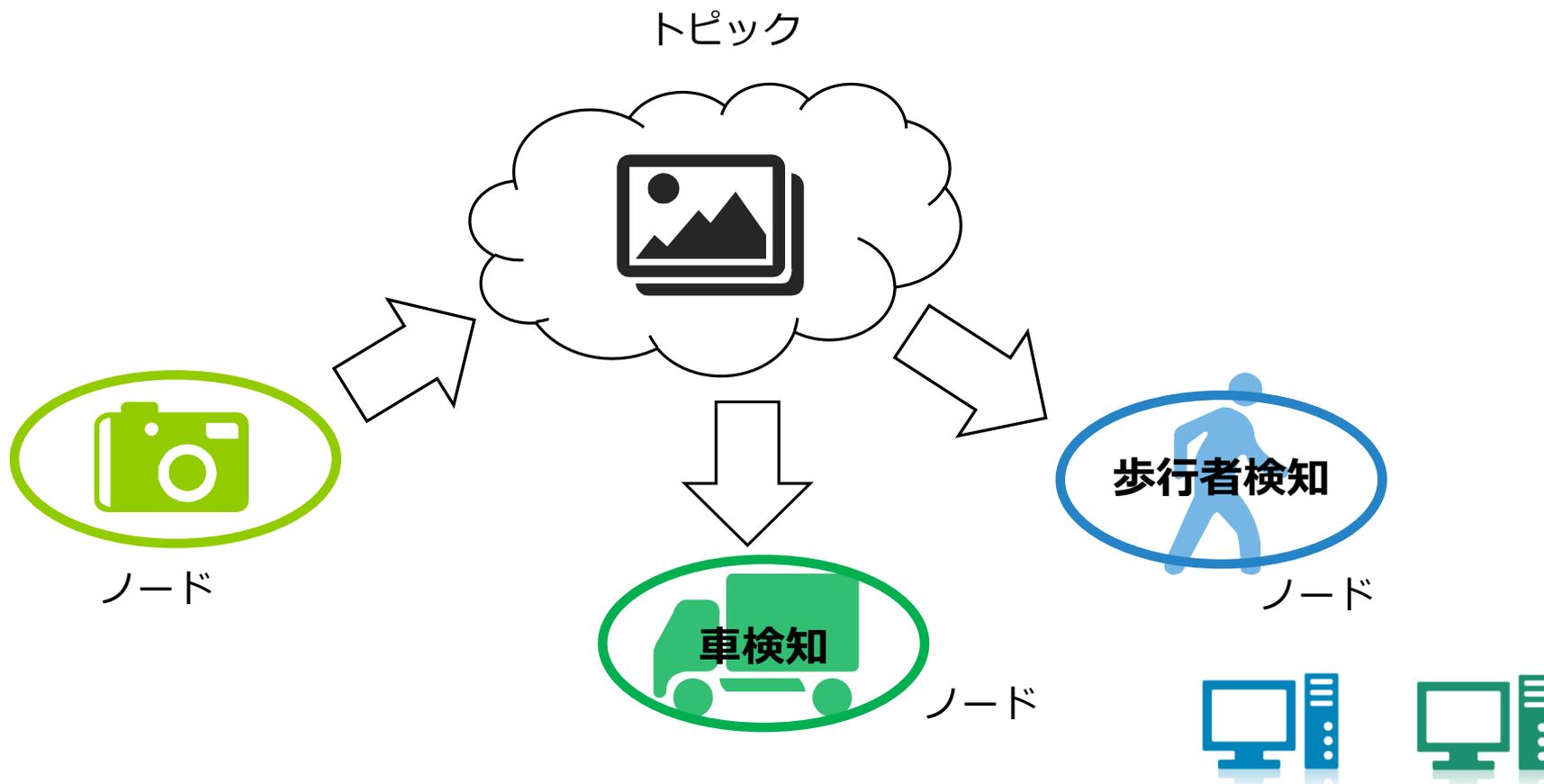
➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離



# Publish / Subscribe モデル

処理を**ノード**として分割・管理し、**トピック**を介してデータのやり取りを行う。

➡ 再利用性・生産性の向上, 分散環境への高い親和性, 障害分離

NEW!

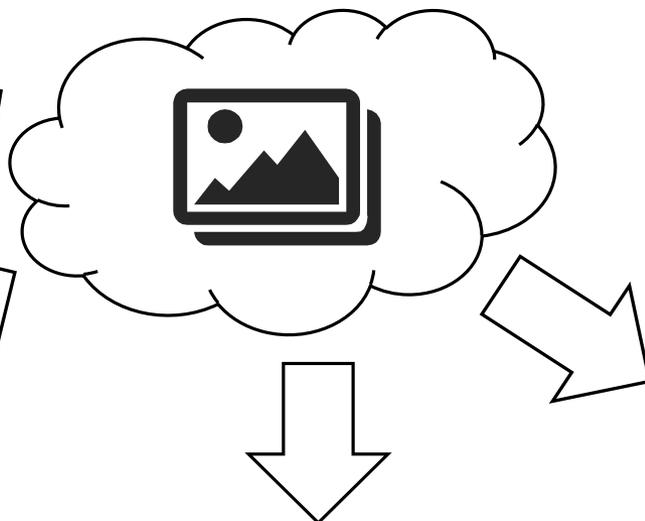


ノード



ノード

トピック



ノード



ノード



# ROS の 特長

## データの保存 : rosbag

実データ（トピック情報）を保存可能

## 視覚化・シミュレーション

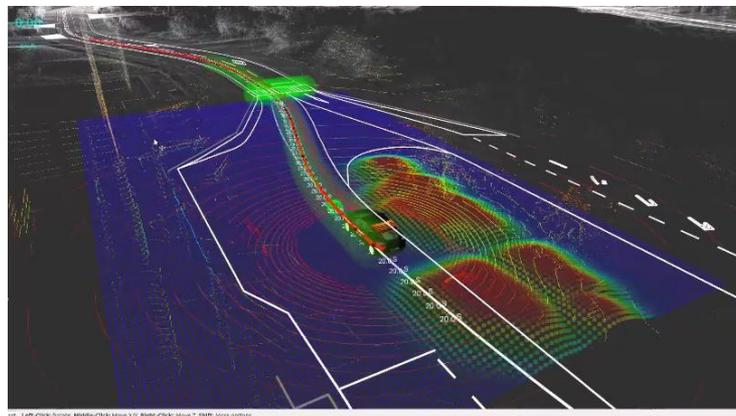


## RViz: 3D視覚化ツール

簡単にシステム状態を視覚化可能

[再生データ]

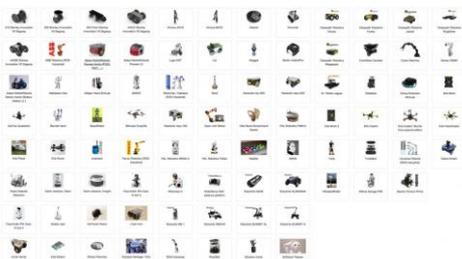
- 記録したセンサデータ（**rosbag**ファイル）
- 指定した値のデータ



# ROS の 特長

## ハードウェア

様々なロボットやセンサをサポート



## パッケージ

2,000を超えるソフトウェアパッケージで効率的開発

## ライブラリ

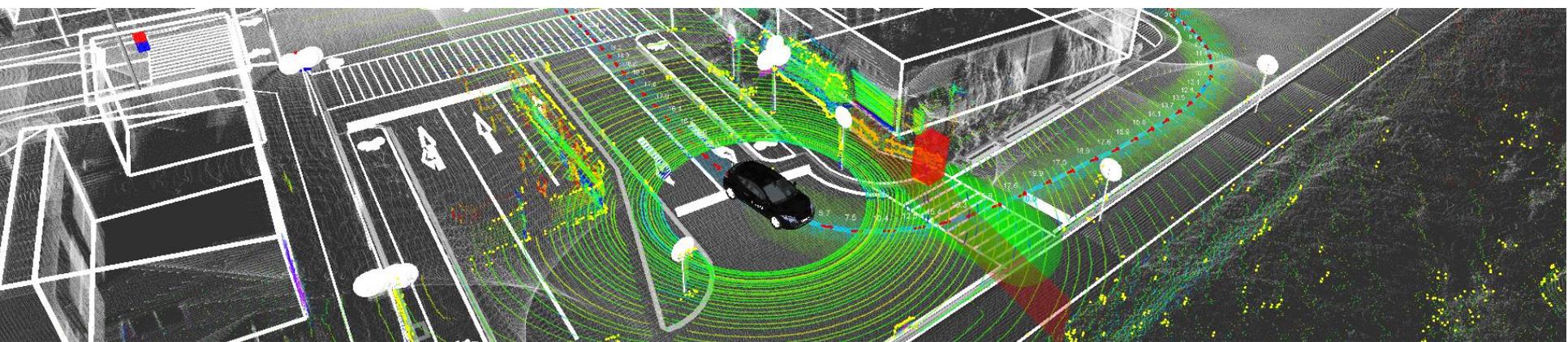
座標変換・画像処理・点群処理  
など豊富にサポート



多くのソフトウェアが  
オープンソースで

**豊富なパッケージ**  
(デバイスドライバやライブラリ)





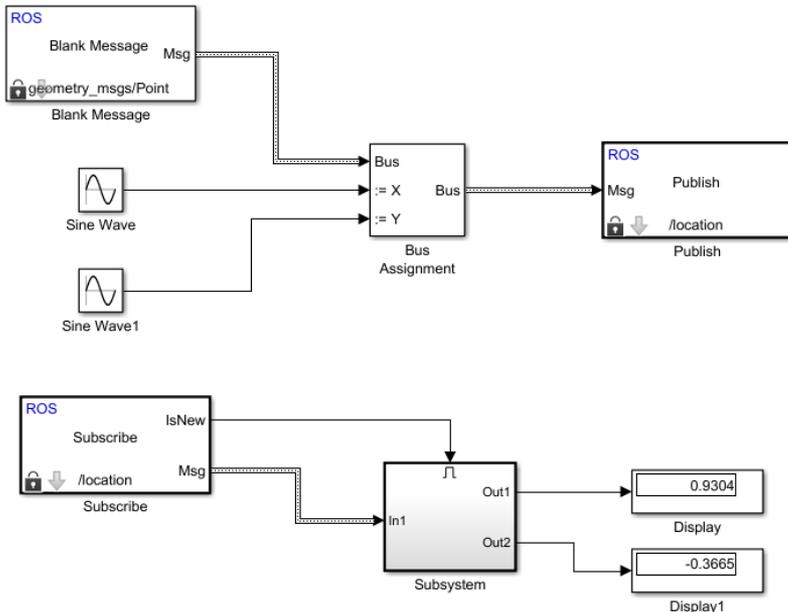
自動運转向けソフトウェアAutowareと

**MATLAB<sup>®</sup> / Simulink<sup>®</sup> の連携**

# MATLAB<sup>®</sup> / Simulink<sup>®</sup>

## MATLAB/Simulink

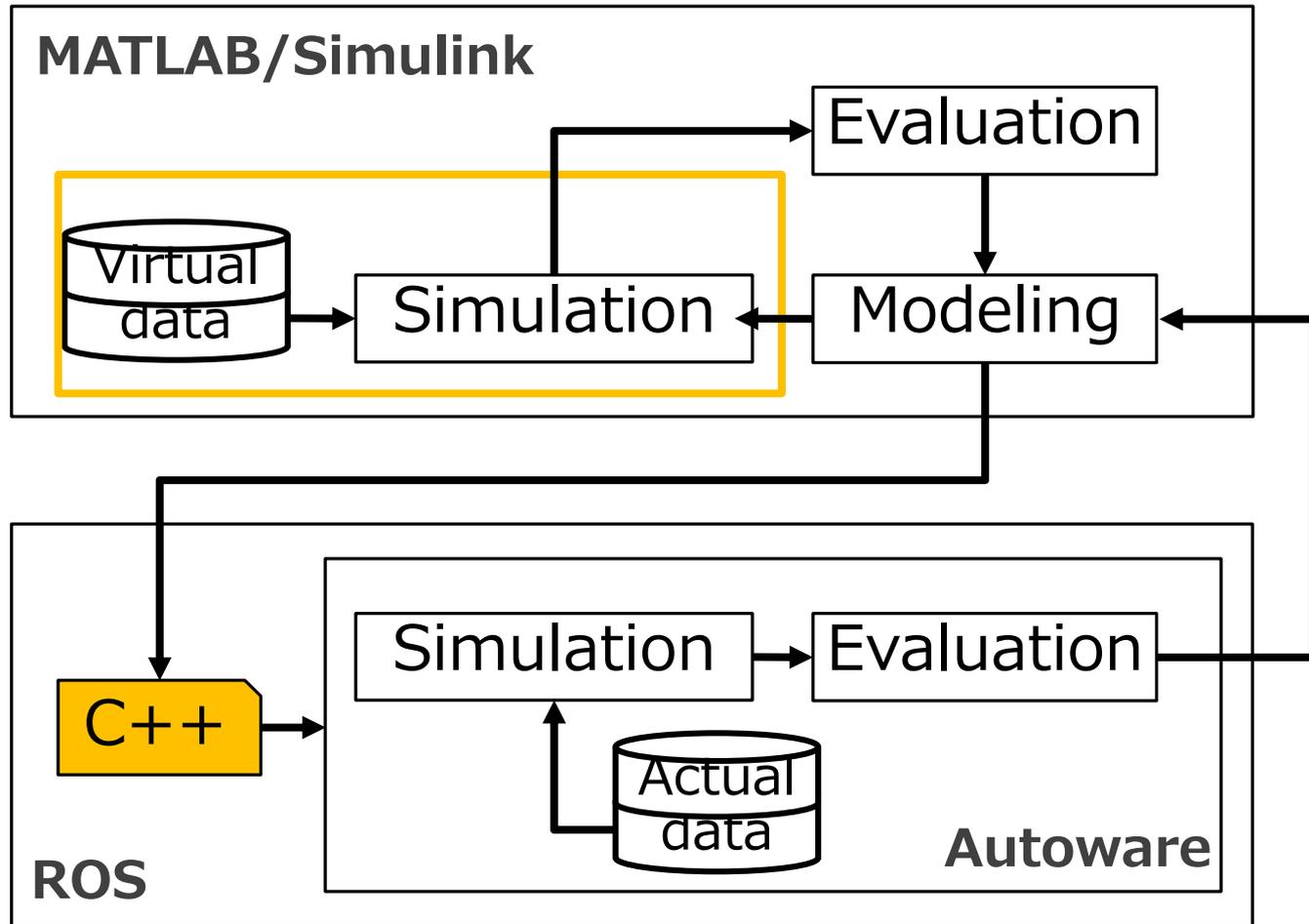
- モデリング、シミュレーション、解析用ソフトウェア
- 提供されている関数を用いることで、モデリングの質・効率を向上させる



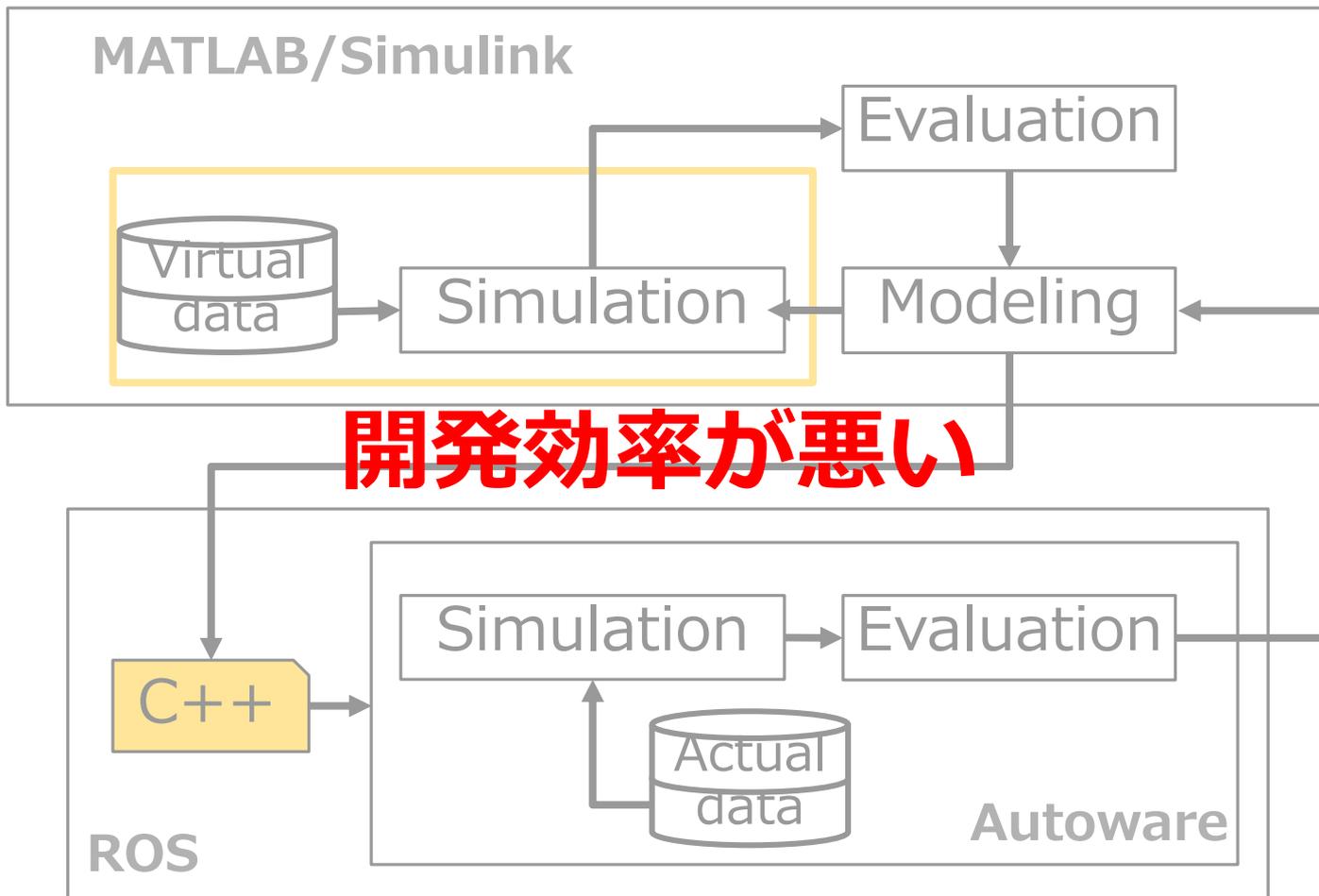
MATLAB R2016b - academic use

```
1 %% 初期化
2 clear;
3 clear global variable;
4 close all;
5 clc;
6
7 %% rosネットワークへの接続
8 rosinit('192.168.1.183')
9
10 %% Nodeの作成
11 node = robotics_ros.Node('lane_select_mat');
12
13 %% 変数の定義
14 global g_pub
15 global g_lane_number
16 g_lane_number = 0;
17 global g_lane_array
18 g_lane_array = rosmessage('waypoint_follower/LaneArray');
19
20 %% Pub/Subの定義
21 config_sub = robotics_ros.Subscriber(node, '/config/lane_select', 'run
22 sub = robotics_ros.Subscriber(node, '/traffic_waypoints_array', 'waypoi
23 g_pub = robotics_ros.Publisher(node, '/base_waypoints', 'waypoint_foll
```

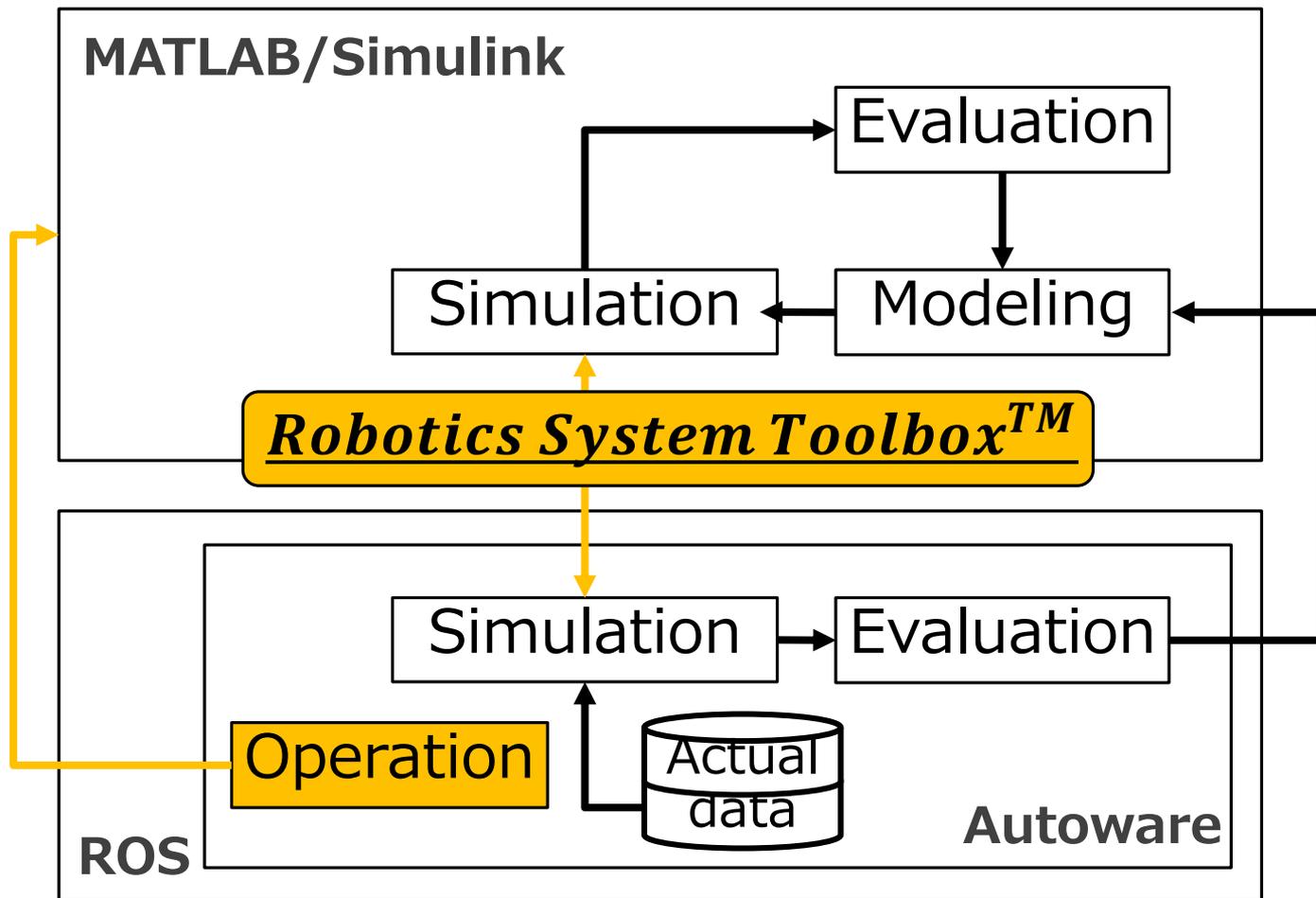
# 現在の開発フレームワーク



# 現在の開発フレームワーク-



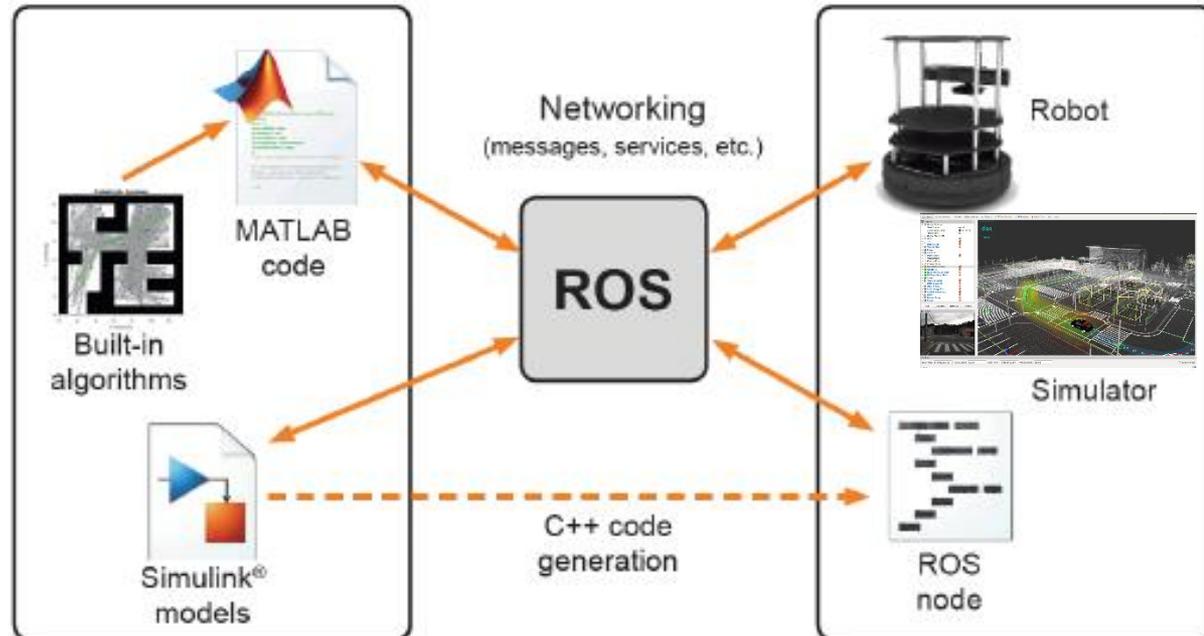
# 提案フレームワーク



# Robotics System Toolbox

## Robotics System Toolbox

- ROSとMATLAB/Simulink間のインタフェースを提供
  - ROSとMATLAB/Simulink間でPublish/Subscribeモデルに基づいたノード間通信が可能



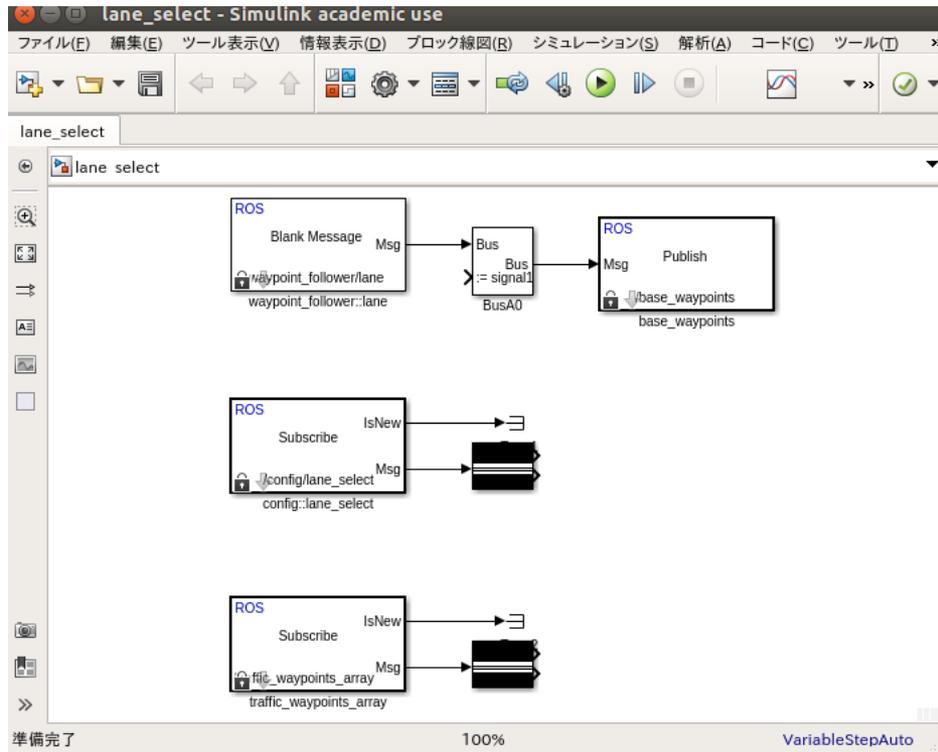
# 提供機能 (1/2)

## MATLAB/Simulink用のテンプレート生成

### MATLAB template script

```
Node_simulink_model_creat.m x lane_select_simulink_model_creat.m x lane_select.m x +
1 node = robotics.ros.Node('lane_select')
2 base_waypoints_pub = robotics.ros.Publisher(node, '/base_waypoints', 'waypoint_follower/
3 runtime_manager_ConfigLaneSelect_sub = robotics.ros.Subscriber(node, '/config/lane_select
4 waypoint_follower_LaneArray_sub = robotics.ros.Subscriber(node, '/traffic_waypoints_arra
5
6 config_lane_select_Callback(~, msg)
7 and
8 traffic_waypoints_array_Callback(~, msg)
9 and
```

### Simulink template model



## MATLAB/Simulink用のRuntime Manager

The screenshot displays the Runtime Manager interface with the following components:

- Navigation Tabs:** Quick Start, Setup, Map, Sensing, Computing, Interface, Database, Simulation, Status, Topics, MATLAB/Simulink.
- MATLAB/Simulink code:** Two input fields for MATLAB code and Simulink code, both pointing to `/home/shota/Autoware/ros/src/util/packages/runtime_manager/scripts`.
- Code generation:** Includes a dropdown for `rqt_graph_aware` and buttons for `MATLAB template code` and `Simulink template model`, with `lane_select` entered in the adjacent field.
- Info Panel:** Shows details for Node `[/lane_select]`, including publications (`/rosout`, `/base_waypoints`) and subscriptions (`/traffic_waypoints_array`, `/config/lane_select`, `/clock`).
- System Resources:** A bar chart showing CPU usage for 8 cores (CPU0-CPU7) and memory usage. CPU0 is at 25.0%, CPU3 and CPU4 at 7.7%, CPU6 at 7.7%, and CPU7 at 8.3%. Memory usage is 2GB/31GB (7%).
- Footer:** The AutoWare logo is visible in the bottom left corner.

# デモンストレーション -経路計画-

The image displays a ROS simulation environment. On the left, the RViz window shows a 3D view of a car labeled 'velodyne' on a road with lane markings. A path plan is visible as a series of colored lines (yellow, green, blue) on the road surface. The RViz interface includes a toolbar with options like 'Interact', 'Move Camera', 'Select', 'Focus Camera', 'Measure', '2D Pose Estimate', '2D Nav Goal', and 'Publish Point'. At the bottom, a 'Time' panel shows ROS Time: 62283.84, ROS Elapsed: 41.93, Wall Time: 1487590578.34, and Wall Elapsed: 91.95. The 'Experimental' checkbox is unchecked, and the frame rate is 17 fps.

On the right, the RQT window shows a 'Node Graph' for the namespace '/base\_link\_to\_localizer'. The graph includes nodes such as 'points\_map\_loader', 'vector\_map\_loader', 'joint\_state\_publisher', 'play', 'voxel\_grid\_filter', 'ndt\_matching', 'vel\_pose\_mux', 'robot\_state\_publisher', 'velodyne\_nodelet', and 'driver\_nodelet'. The 'Runtime Manager' window below shows a rosbag file being played at a rate of 0.6. The rosbag file path is '/home/shota/.autoware/toyota/autoware-20160905165640.rosbag.bag'. The playback progress is at 30%. The rosbag file details are as follows:

- path: /home/shota/.autoware/toyota/autoware-20160905165640.rosbag.bag
- version: 2.0
- duration: 2:52s (172s)
- start: Sep 05 2016 16:57:11.94 (1473062231.94)
- end: Sep 05 2016 17:00:04.30 (1473062404.30)
- size: 10.9 GB
- messages: 20515
- compression: none [2520/2520 chunks]
- types: sensor\_msgs/Image [060021388200f6f0f447d0fcd9c64743]
- vehicle\_socket/CanInfo [605f4356821f92a8fec1a756259316df]
- velodyne\_msgs/VelodyneScan [50804fc9533a0e579e632c04ae70566]
- topics: /can\_info 16318 msgs : vehicle\_socket/CanInfo
- /image\_raw 2519 msgs : sensor\_msgs/Image
- velodyne\_msgs/VelodyneScan 1678 msgs : velodyne\_msgs/VelodyneScan

At the bottom right, a system resource monitor shows CPU usage for CPU0 (30.8%), CPU1 (41.7%), CPU2 (50.0%), CPU3 (40.0%), CPU4 (81.8%), CPU5 (50.0%), CPU6 (27.3%), and CPU7 (30.8%). Memory usage is 3GB/31GB (12%). The AutoWare logo is visible at the bottom center.