

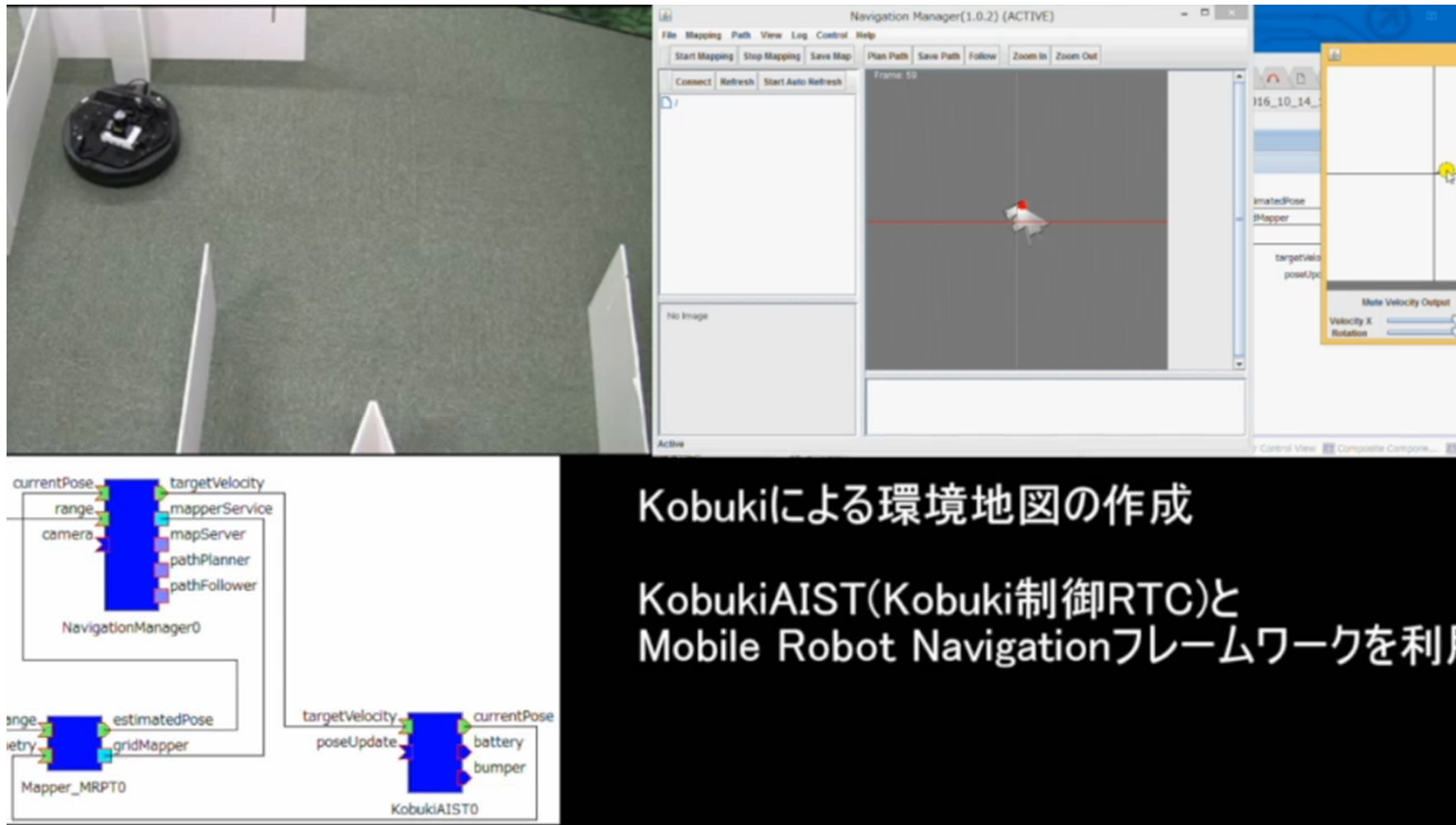
2018年3月12日（月）  
OSSユーザーのための勉強会



# ロボット開発/ RT-Middleware の 特長と最新動向

国立研究開発法人産業技術総合研究所  
ロボットイノベーション研究センター  
ロボットソフトウェアプラットフォーム研究チーム長  
安藤 慶昭

# RTミドルウェアとは？



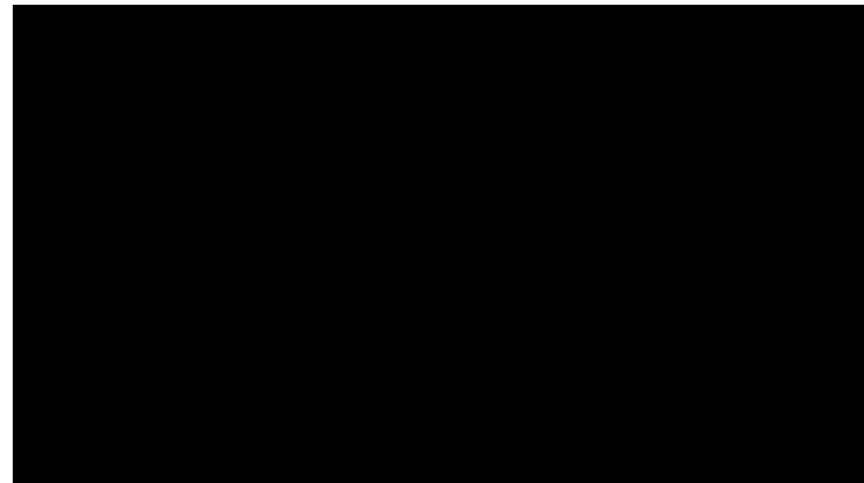
The top left shows a physical mobile robot in a simulated environment. The top right is a screenshot of the 'Navigation Manager(1.0.2) (ACTIVE)' software interface, which includes a map view, a list of modules, and control parameters like 'Velocity X' and 'Rotation'. The bottom left is a block diagram showing the system architecture. It includes a 'NavigationManager0' block with inputs like 'currentPose', 'range', and 'camera', and outputs like 'targetVelocity', 'mapperService', 'mapServer', 'pathPlanner', and 'pathFollower'. Below it is a 'Mapper\_MRPT0' block with inputs 'range' and 'battery', and outputs 'estimatedPose' and 'gridMapper'. To the right is a 'KobukiAIST0' block with inputs 'targetVelocity' and 'poseUpdate', and outputs 'currentPose', 'battery', and 'bumper'.

Kobukiによる環境地図の作成

KobukiAIST(Kobuki制御RTC)と  
Mobile Robot Navigationフレームワークを利用

モジュール化されたソフトウェアを組み合わせ  
システムを構成するミドルウェア・ツール群

# RTミドルウェアとは？



主に多自由度（モータ数が多い）のロボットの  
リアルタイム制御などに力を発揮

# はじめに

- RTミドルウェアの概要
  - 基本概念
  - OSSにまつわる問題
- ROSとの比較、動向
  - ROS2, ROS-industrial
- オープンソース活用プロジェクト
  - ロボット新戦略と新規NEDOプロジェクト
  - RRI調査検討委員会
- まとめ

# RTとは?

- RT = Robot Technology      cf. IT
  - ≠Real-time
  - 単体のロボットだけでなく、さまざまなロボット技術に基づく機能要素をも含む (センサ、アクチュエータ, 制御スキーム、アルゴリズム、etc….)

産総研版RTミドルウェア

# OpenRTM-aist

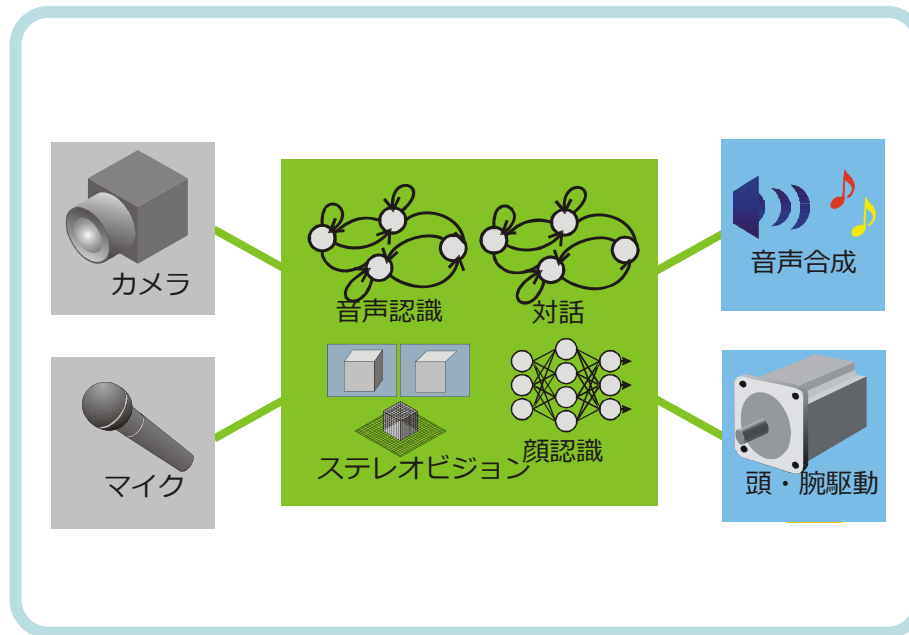
- RT-Middleware (RTM)
  - RT要素のインテグレーションのためのミドルウェア
- RT-Component (RTC)
  - RT-Middlewareにおけるソフトウェアの基本単位

# ロボットソフトウェア開発の方向

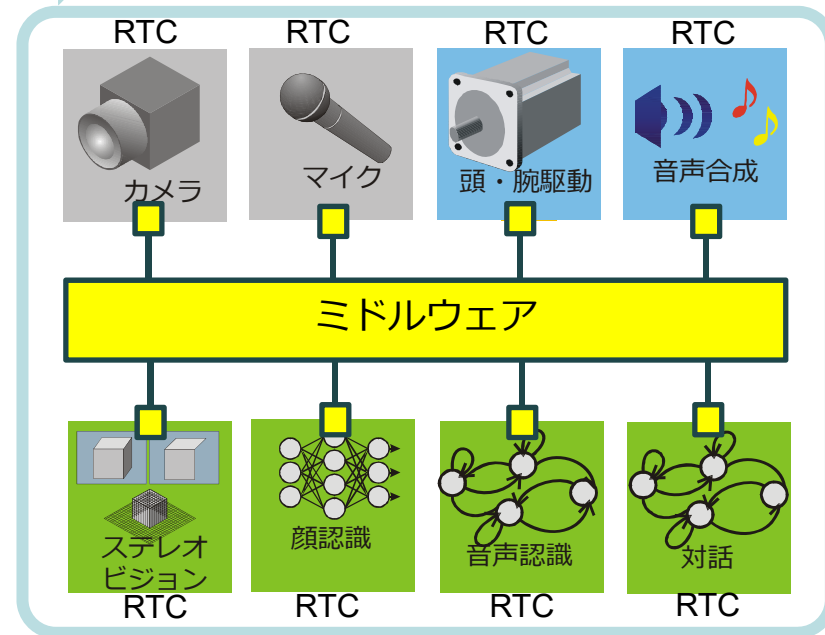
従来型開発



コンポーネント指向開発



- ✓ 様々な機能を融合的に設計
- ✓ 実行時の効率が高いが、柔軟性に欠ける
- ✓ システムが複雑化してくると開発が困難に



- ✓ 大規模複雑な機能の分割・統合
- ✓ 開発・保守効率化（機能の再利用等）
- ✓ システムの柔軟性向上

2000年ごろからロボットミドルウェア・ロボットOSの開発が活発化  
新規ロボット開発にミドルウェア利用することが主流に

# RTコンポーネント化のメリット

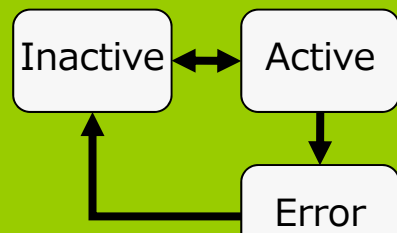
モジュール化のメリットに加えて

- ソフトウェアパターンを提供
  - ロボットに特有のソフトウェアパターンを提供することで、体系的なシステム構築が可能
- フレームワークの提供
  - フレームワークが提供されているので、コアのロジックに集中できる
- 分散ミドルウェア
  - ロボット体内LANやネットワークロボットなど、分散システムを容易に構築可能

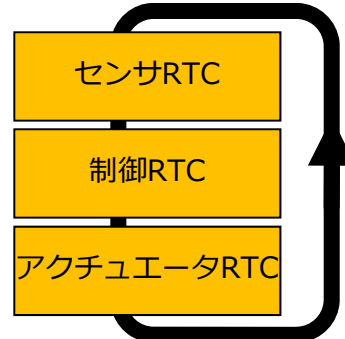
# RTコンポーネントの主な機能

## アクティビティ・実行コンテキスト

共通の状態遷移



複合実行

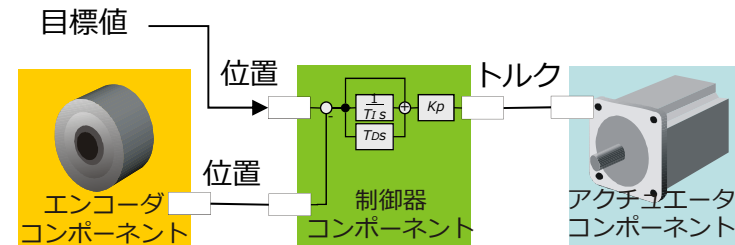


ライフサイクルの管理・コアロジックの実行

## データポート

- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断

サーボの例



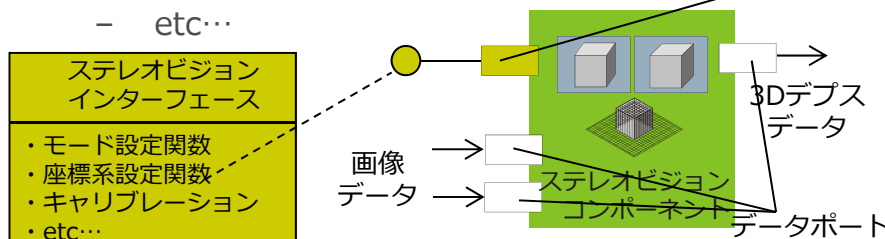
データ指向通信機能

## サービスポート

- 定義可能なインターフェースを持つ
- 内部の詳細な機能にアクセス
  - パラメータ取得・設定
  - モード切替
  - etc...

ステレオビジョンの例

サービスポート

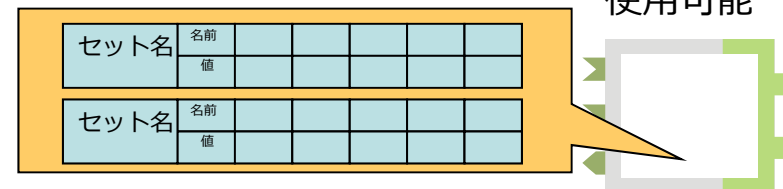


サービス指向相互作用機能

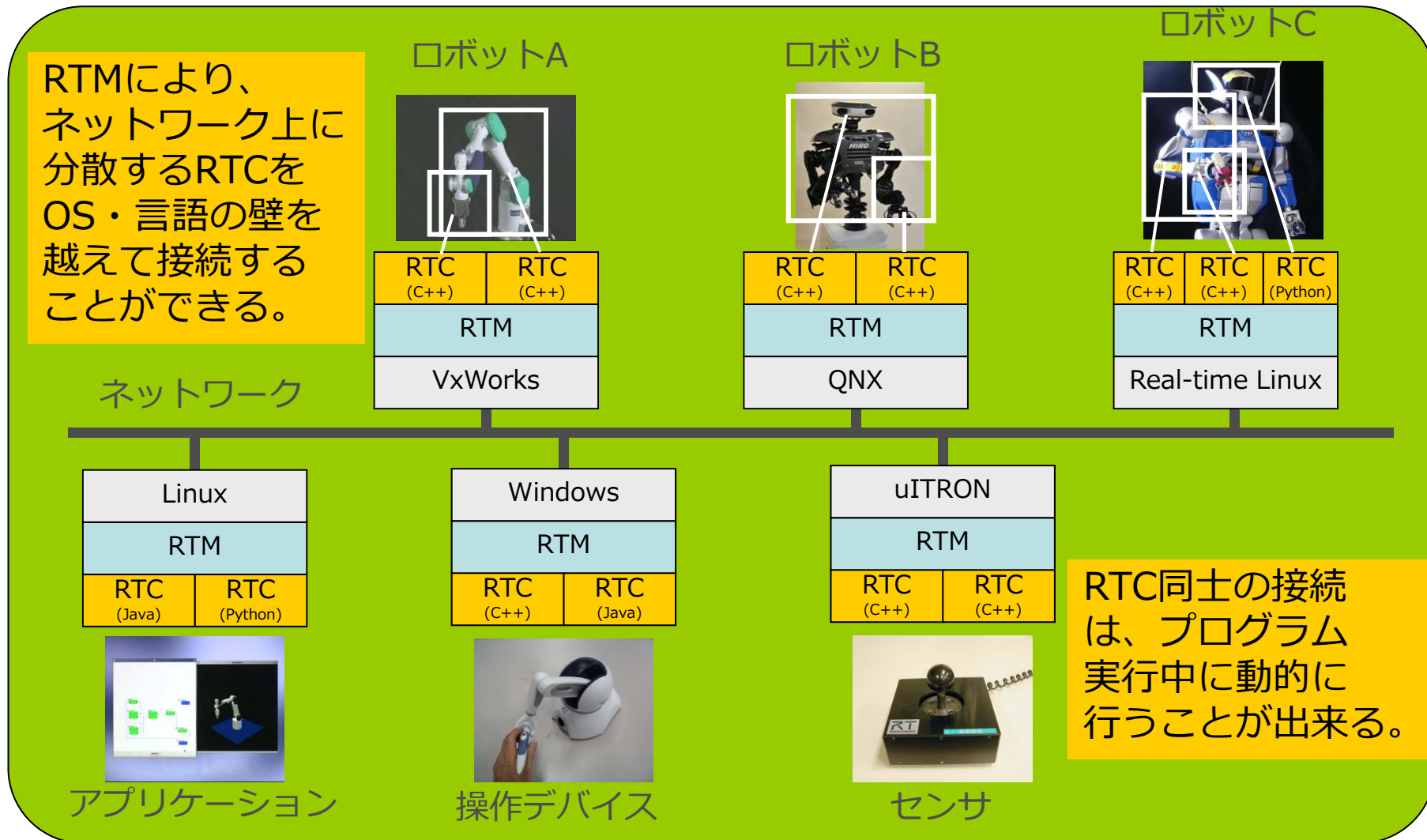
## コンフィギュレーション

- パラメータを保持する仕組み
- いくつかのセットを保持可能
- 実行時に動的に変更可能

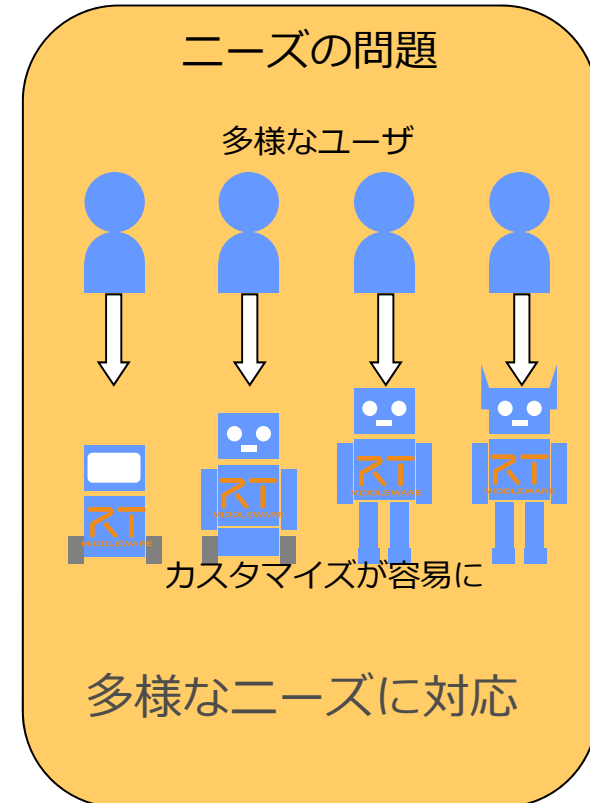
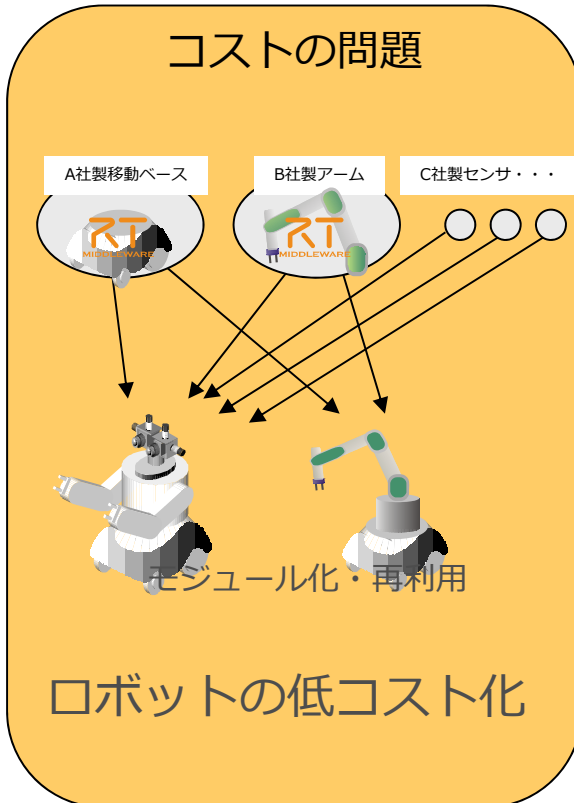
複数のセットを  
動作時に  
切り替えて  
使用可能



# RTミドルウェアによる分散システム



# RTミドルウェアの目的 モジュール化による問題解決



ロボットシステムインテグレーションによるイノベーション

# OpenRTM-aistのライセンス

OpenRTM-aist (C++, Python, Java, ツール群)

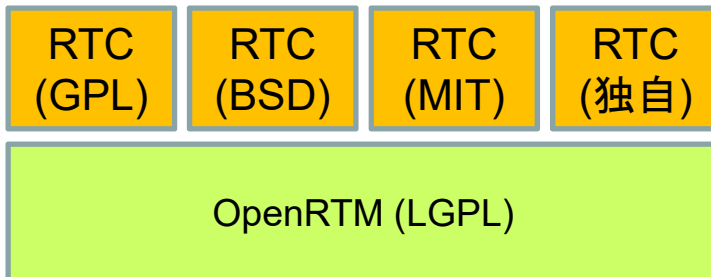
||

LGPL (ツールはEPL)と個別契約のデュアルライセンス方式

EPL: Eclipse Public License (≒LGPLに特許条項を付加したライセンス)

※特許を含むコードを寄贈しても後から特許権を主張できない

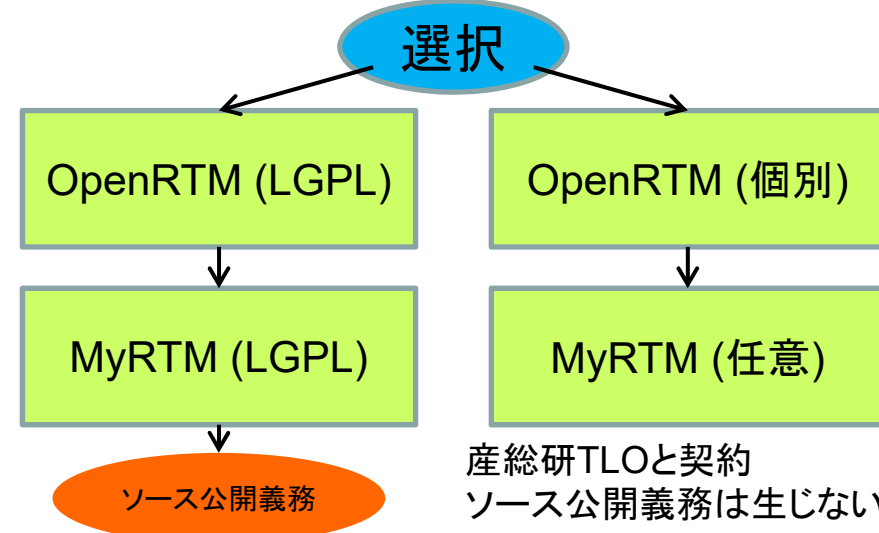
## RTコンポーネントを作る場合



RTコンポーネントは動的リンクライブラリ形式で提供可能なので、ライセンスは自由に設定可能。商用ライセンスを設定して販売することもできる。

RTCの自由な流通を保証

## 新たなRTMやツールを作る場合



ハード固有の改良に対して  
クローズド戦略も保証

# RTC国際標準

Date: September 2012



## Robotic Technology Component (RTC)

Version 1.1

Normative reference: <http://www.omg.org/spec/RTC/1.1>  
 Machine consumable files: <http://www.omg.org/spec/RTC/20111205/>  
 Normative:  
<http://www.omg.org/spec/RTC/20111205/rtc.xml>  
<http://www.omg.org/spec/RTC/20111205/rtc.h>  
<http://www.omg.org/spec/RTC/20111205/rtc.idl>  
 Non-normative:  
<http://www.omg.org/spec/RTC/20111205/rtc.eap>

### 標準化履歴

- 2005年9月  
Request for Proposal 発行(標準化開始)
- 2006年9月  
OMGで承認、事実上の国際標準獲得
- 2008年4月  
OMG RTC標準仕様 ver.1.0公式リリース
- 2012年9月  
ver. 1.1改定
- 2015年9月  
FSM4RTC(FSM型RTCとデータポート標準) 採択

## OMG国際標準

標準化組織で手続きに沿って策定

- 1組織では勝手に改変できない安心感
- 多くの互換実装ができつつある
- 競争と相互運用性が促進される

RTミドルウェア互換実装は10種類以上

| 名称             | ベンダ            | 特徴                               | 互換性 |
|----------------|----------------|----------------------------------|-----|
| OpenRTM-aist   | 産総研            | NEDO PJで開発。参照実装。                 | --- |
| HRTM           | ホンダ            | アシモはHRTMへ移行中                     | ◎   |
| OpenRTM.NET    | セック            | .NET(C#, VB, C++/CLI, F#, etc..) | ◎   |
| RTM on Android | セック            | Android版RTミドルウェア                 | ◎   |
| RTC-Lite       | 産総研            | PIC, dsPIC上の実装                   | ○   |
| Mini/MicorRTC  | SEC            | NEDOオープンイノベーションPJで開発             | ○   |
| RTMSafety      | SEC/AIST       | NEDO知能化PJで開発・機能安全認証取得            | ○   |
| RTC CANOpen    | SIT, CiA       | CAN業界RTM標準                       | ○   |
| PALRO          | 富士ソフト          | 小型ヒューマノイドのためのC++ PSM 実装          | ×   |
| OPRoS          | ETRI           | 韓国国家プロジェクトでの実装                   | ×   |
| GostaiRTC      | GOSTAI, THALES | ロボット言語上で動作するC++ PSM 実装           | ×   |

特定のベンダが撤退しても  
ユーザは使い続けることが可能

# 実用例・製品化例



HRPシリーズ: 川田工業、AIST

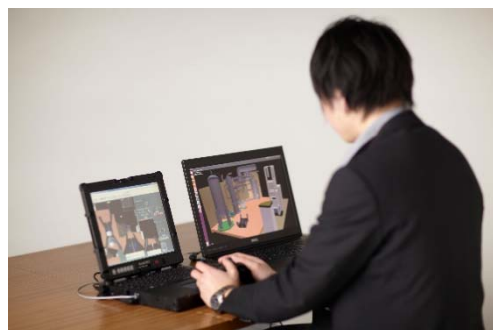


S-ONE : SCHAFT



DAQ-Middleware: KEK/J-PARC

KEK: High Energy Accelerator Research Organization  
J-PARC: Japan Proton Accelerator Research Complex



災害対応ロボット操縦シミュレータ:  
NEDO/千葉工大



HIRO, NEXTAGE open: Kawada Robotics



RAPUDA : Life Robotics



ビュートロバーRTC/RTC-BT(VSTONE)



OROCHI (アールティ)



新日本電工他: Mobile SEM

# プロジェクトページ

- ユーザが自分のRTCを登録
  - オープンソースのものは自由にダウンロード可能
- 
- 品質管理・保守が課題
    - 過去のプロジェクトでは、ソフト専門企業が検証・ドキュメント整備・保守を実施したことも
  - ライセンスの問題
    - 大学などのRTCはライセンスが明記されていないものも
    - RTCBuilderではライセンス記入欄を追加
  - 特許の問題
    - 個々のRTCの特許調査は事実上不可能
    - 過去のプロジェクトでは予防調査は実施

The screenshot displays the OpenRTM-aist website interface. The top navigation bar includes links for Home, Downloads, Documents, Community, Projects, Hardware, and Pukiwiki Manual. The main content area shows two project listings. The first project, 'Kartoライブラリを用いた自律地図生成システム' by yusuke hayashi, includes details about its purpose (autonomous mapping using Karto), its license (MIT), and a list of dependencies (numpym, opencv, PyRTSocm). The second project, '対戦アルゴリズムを容易に変更可能なエアホッケーロボットRTC' by uyaponz, describes a system for playing air hockey with a robot and lists its dependencies (opencv, numpy, PyRTSocm). Both project entries have a '続きを見る' (View more) button.

現在 400弱のプロジェクト登録数

# サマーキャンプ

- 毎年夏に1週間開催
- 今年：7月30日～8月3日
- 募集人数：20名
- 場所：産総研つくばセンター
- 座学と実習を1週間行い、最後にそれぞれが成果を発表
- 産総研内のさくら館に宿泊しながら夜通し？コーディングを行う！



# RTミドルウェアコンテスト

- SICE SI（計測自動制御学会 システムインテグレーション部門講演会）のセッションとして開催
  - － 各種奨励賞・審査基準開示:5月頃
  - － エントリー〆切：8月21日(SI2017締切)
  - － 講演原稿〆切：9月25日
  - － ソフトウェア登録：10月ごろ
  - － オンライン審査：11月下旬～
  - － 発表・授賞式：12月ごろ
- 2016年度実績
  - － 応募数：13件
  - － 計測自動制御学会学会RTミドルウェア賞（副賞10万円）
  - － 奨励賞（賞品協賛）：2件
  - － 奨励賞（団体協賛）：8件
  - － 奨励賞（個人協賛）：5件
- 詳細はWebページ：[openrtm.org](http://openrtm.org)
  - － コミュニティ→イベント をご覧ください



# ロボットにおけるオープンソースソフトウェアの利用

- ソフトウェアの大規模化に対応するには必要
  - コモディティー化した領域で利用
  - いまや、プランニング、SLAMもコモディティー化

## メリット

- 世界の研究者・開発者の最新技術が利用可能
- 競争領域へのリソース集中可能
- アルゴリズムの検討・検証・修正・改変が可能
- コミュニティの力を借りることが可能

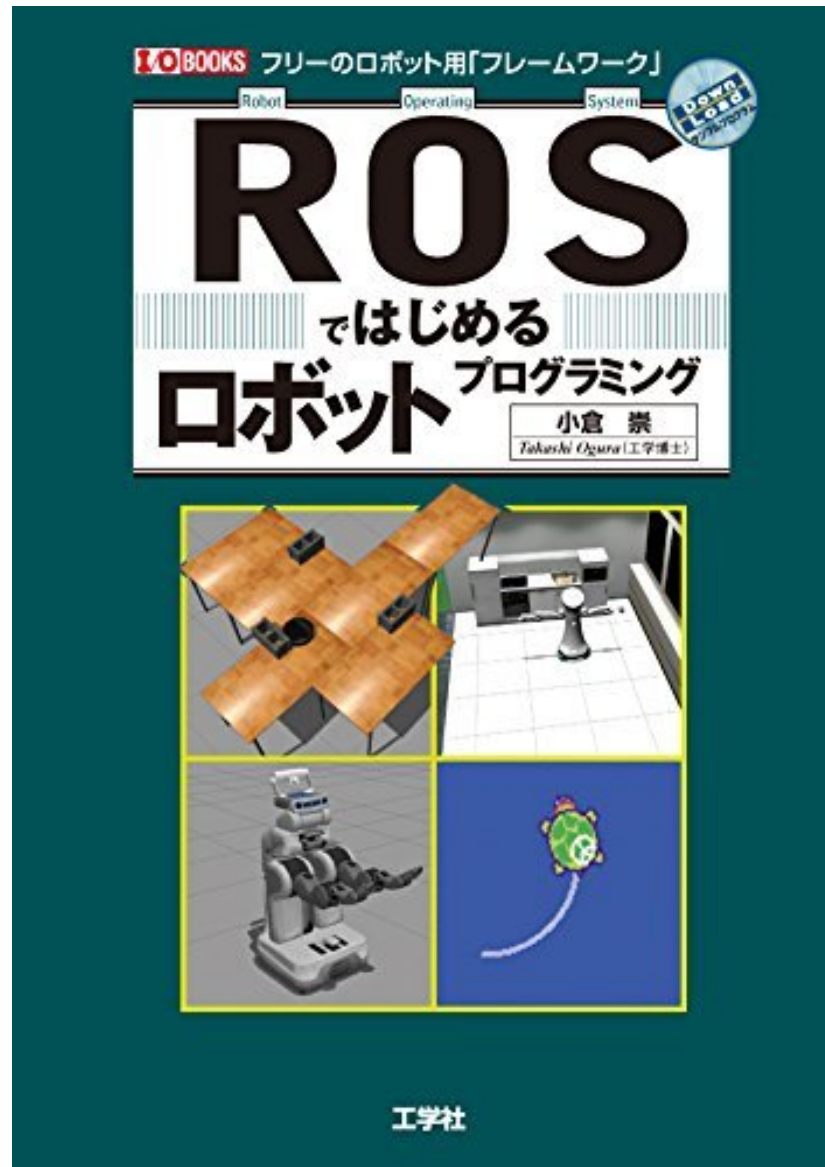
## デメリット

- 製品化時のライセンス問題
- 特許は依然独自に調査が必要
- 品質管理を自社で行う必要性
- 安全関連系には使いづらい
- 第3者による改変・バージョンアップ・仕様変更への対応が必要

ロボット業界でもメリットは認識されつつあるも  
デメリットを克服できない場合、やはり独自実装になりがち

# ROS (&ROS2、 ROS-I)

# ROSとRTM



# ROSとRTM

## ROSの特長

- UNIX文化に根差した設計思想
  - Windows等Linux以外は原則サポートしない
  - ROS2ではLinux以外もサポート
- 独自のパッケージ管理システムを持つ
- ノード（コンポーネント）の質、量ともに豊富
  - OSRFが直接品質を管理しているノードが多数
- ユーザ数が多い
- メーリングリストなどの議論がオープンで活発
  - コアライブラリの仕様が固定しにくい
- ドキュメントが豊富

## OpenRTMの特長

- 対応OS・言語の種類が多い
  - Windows、UNIX、uITRON、T-Kernel、VxWorks、QNX
  - Windowsでネイティブ動作する
- 日本国内がメイン
  - 日本語ドキュメント・ML・講習会
  - ユーザ数が少ない
- GUIツールがあるので初心者向き
  - コマンドラインツールの種類は少ない
  - rtshell
- 仕様が標準化されている
  - OMGに参加すればだれでも変更可
  - サードパーティー実装が作りやすい
  - すでに10程度の実装あり
- コンポーネントモデルが明確
  - オブジェクト指向、UML・SysMLとの相性が良い
  - モデルベース開発
- IEC61508機能安全認証取得
  - RTMSafety

# ROS1→ROS2へ

## OSRF内部の人のコメント

- NASAの仕事を請け負った時に、独自形式のROS messageはNASAでは使えないから、プロトタイプをROSで実装後にすべて作り直した
- NASAでは何らかの標準に準拠したものでないと使えない。その時は結局DDSを使用した。
- それ以外にもROS1では、1ノード1プロセス、コンポーネントモデルがないので、モデルベース開発にならない、ROS masterがSPOFになっているなど不都合な点が多々ある
- それ故、ROS2ではこれらの問題点を克服するため全く新しい実装にする。

# ROS2

- 最新版：2017年12月 Ardent Apaloneリリース
- 標準ミドルウェアの利用
  - 自前主義からの脱却（NASAや商用システムでは何らかの標準標準が求められるため）
  - 通信部分はOMG標準のDDS（Data Distribution Service、OMG規格）を採用
- コンポーネントモデルを導入することにした
  - RTMのように組み込み、性能を意識したアーキテクチャへ
- 対応OSの拡大
  - これまでは、ある特定のLinux（Ubuntu Linux）のみ
  - ROS2では、Windows、Macにも対応
  - ただし、リアルタイムOS（VxWorks、QNX等）への対応はなし

OMGによる通信規格DDSを採用



航空・軍事・医療・鉄道などで実績のある通信ミドルウェア標準

[http://design.ros2.org/articles/ros\\_middleware\\_interface.html](http://design.ros2.org/articles/ros_middleware_interface.html)

# ROS-I (industrial)

- 産業用にROSを活用するためのコンソーシアムでありオープンソースプロジェクト
  - [https://www.youtube.com/watch?v=xenFvis\\_iVc](https://www.youtube.com/watch?v=xenFvis_iVc)
  - ROS-I America
  - ROS-I Europe
  - ROS-I Asia Pacific

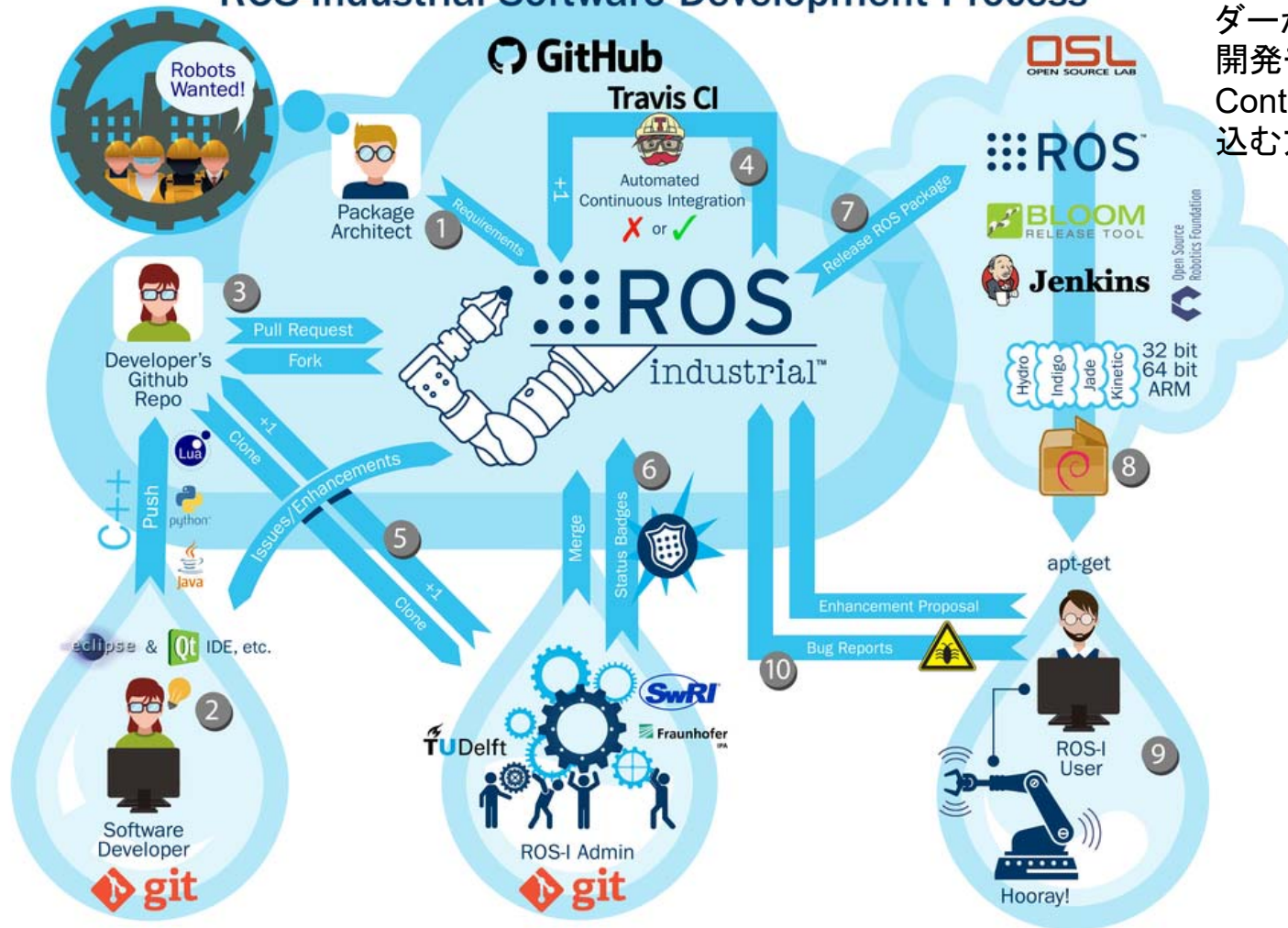


Amazon Picking Challenge  
優勝したTU Delft は ROS-Iを利用

産業利用のためのソフトウェアの「品質管理」および  
機能拡張・新規開発を目的としたコンソーシアム

# ROS-I 開発プロセス

## ROS-Industrial Software Development Process

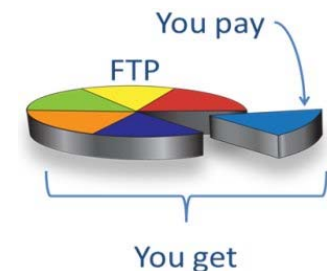


基本的には複数のステークホルダーが関与するオープンソース開発モデルに従う。Contributor は排除せずに、取り込む方針。

# Focused Technical Projects (FTP)

- Fullメンバーが参加可能
- 新規機能の開発を実施可能
- メンバーが集まり資金がcritical mass に到達したらPJ開始
- IPの扱い: 基本的にはオープンにするが、FTPメンバーがコードの公開を(最大2年間)遅らせることができる
- ただし、参加企業名、詳細な要求仕様、ドキュメント、固有の3Dモデル、図・絵、デモやそのビデオについては秘匿することは可能
- SwRIや国から支援を受けているコードは基本オープンにしなければならない

オープンソースとクローズドソースの  
折り合いをどのようにつけるかが課題



# ロボットへのOSS活用の取り組み NEDO市場化プロジェクト

「オープンロボットソフトウェアのプラットフォーム化技術開発」

# ソフトウェアのプラットフォーム化 によるロボット導入コスト削減

※OSS=オープンソースソフトウェア

## 4. SI費用

ティーチング、チューニング等顧客の要望に応じた作業  
知能化研究・パッケージ化を進めPF化することで削減

## 2. 分野・機能別ソフトウェア

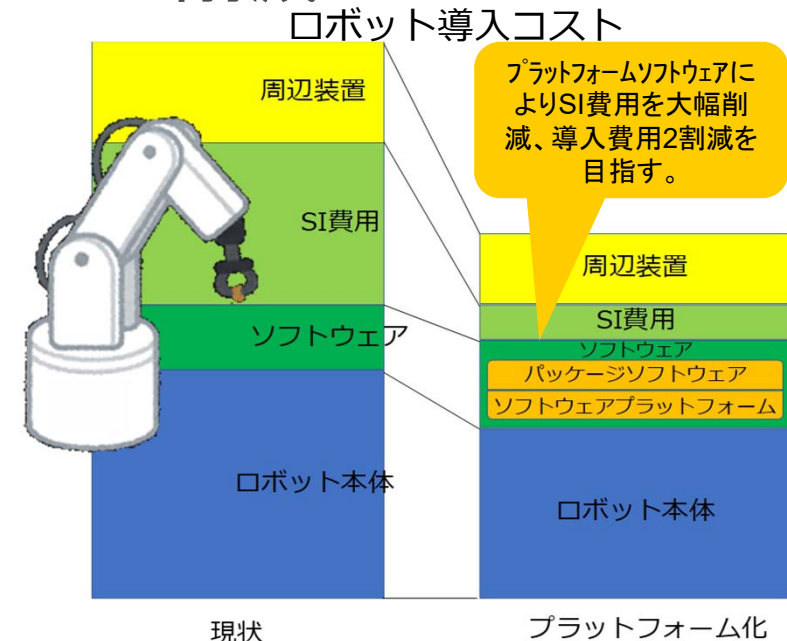
ピックアップ・プレース、SLAM等用途別基本機能  
MoveIt!, OepnCV等OSS活用or機能強化

## 1. ソフトウェア基盤

利用するHW・用途に応じて既存OS利用、  
RTM・ROS等OSS活用or機能強化

## 3. 非機能要件

- ・テスト・検証・性能指標
- ・ドキュメンテーション
- ・教育・コミュニティー
- ・安全規格とのマッチング
- ・ライセンス・特許
- ・長期的な保守・運用体制



分野横断的機能開発

分野別機能開発

横展開可能な技術

市場が成熟するまではコンソーシアム方式が適当

多様な参加者が必要

1社のみの開発は負担大

情報共有の場が必要

# オーブンプラットフォーム戦略

# NEDO プラットフォームロボットPJ (H29年度～) (ロボット活用型市場化適用技術開発PJ)

ロボット革命イニシアティブ協議会 (RRI)  
での議論の結果を提言

資料3

「ロボット活用型市場化適用技術開発プロジェクト」に係る公募要領  
【委託事業】

## ロボット活用型市場化適用技術開発プロジェクト

平成29年度予算案額 17.5億円 (15.0億円)

製造産業局 ロボット政策室  
03-3501-1049

### 事業の内容

**事業目的・概要**

- 「ロボット新戦略」(平成27年2月10日 日本経済再生本部決定)に基づき、ものづくり分野とサービス分野において、これまで実現が困難であった組立工程における柔軟物把持等が可能となるロボットの技術開発を実施します。
- 加えて、小型汎用ロボットの導入コストの2割削減に向け、小型汎用ロボットの本体価格を引き下げ、汎用的な作業・工程に使える小型汎用ロボット(プラットフォームロボット)の開発(ハードウェア・ソフトウェアの共通化)を行います。

**成果目標**

- 平成27年度から平成31年度までの5年間の事業であり、平成31年度までに、上市可能なレベルのロボットを25種類以上開発することを目指します。

**条件(対象者、対象行為、補助率等)**

国

→

(研)新エネルギー・産業技術総合開発機構(NEDO)

→

民間企業等

委託/補助  
(中小企業: 2/3)  
(大企業: 1/2)

### 事業イメージ

**ものづくり分野**  
組立工程における柔軟物把持

整列されていない部品のピッキング工程のロボット化

**サービス分野**  
多様な品物の仕分け

多種多様なサイズ・形状の品物の積付作業のロボット化

**プラットフォームロボット**

| 現状              | 開発後                             |
|-----------------|---------------------------------|
| システムインテグレーション費用 | 共通化によりシステムインテグレーション費用及び本体費用等を削減 |
| ソフトウェア          | システムインテグレーション費用                 |
| 周辺装置            | ソフトウェア                          |
| ロボット本体          | 周辺装置                            |
|                 | ロボット本体                          |

汎用的な用途に活用できるプラットフォームロボットを開発(ハードウェア、ソフトウェアの機能要素を共通化)することで、ロボットシステムの導入コストの削減を図る。(※多種多様なロボットや周辺装置を組み合わせ、最適な自動化システムを構築すること。)

NEDO委託事業として  
2017年度より開始

平成29年5月

国立研究開発法人新エネルギー・産業技術総合開発機構

ロボット・AI部

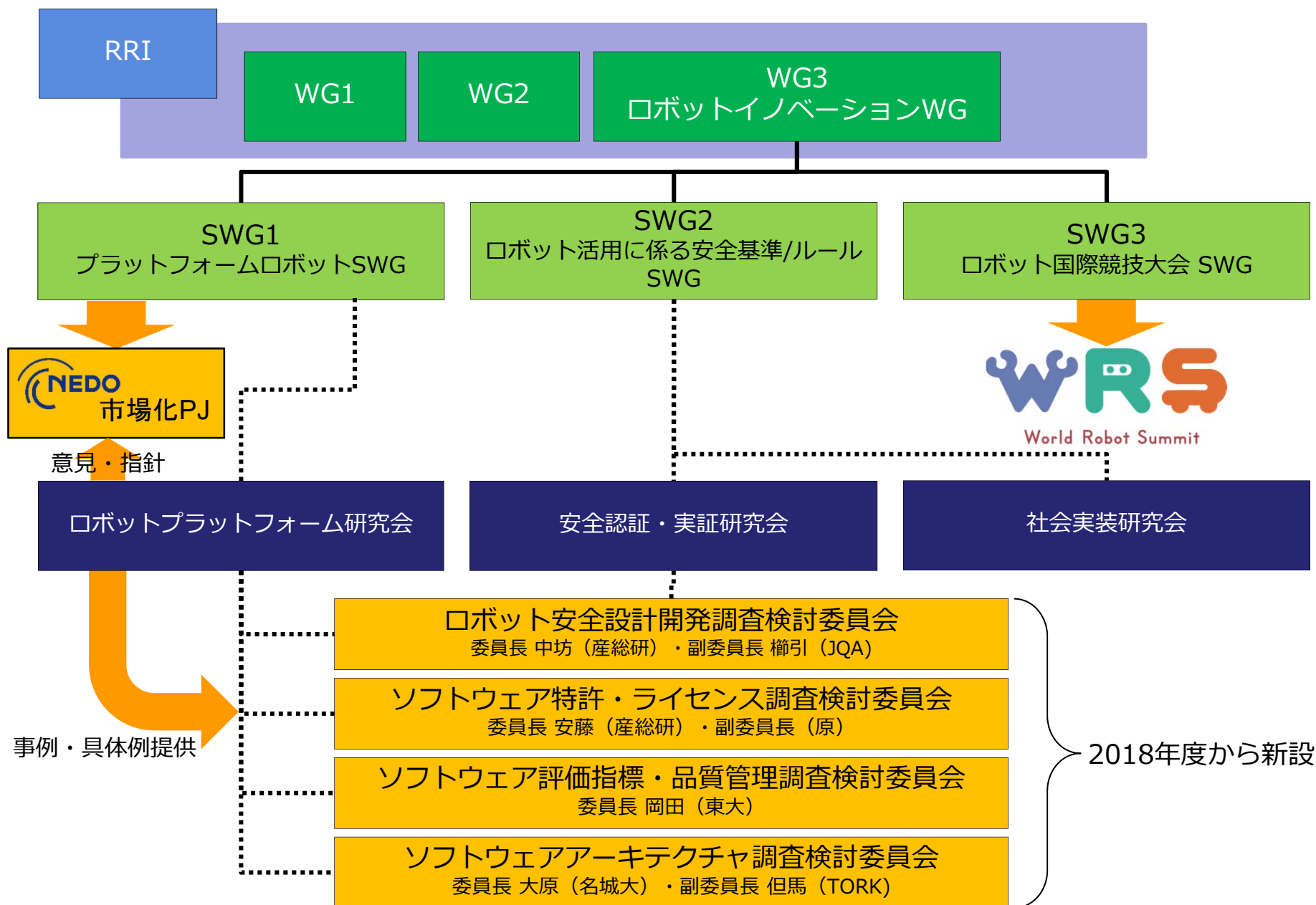
1

### 基本計画に明記

「NEDO事業で開発され標準となっている**RTミドルウェア**(OMG formal/08-04-01)又はデファクト標準となっている**ROS**(Robot Operating System)などの**利用を推奨**する。」

# NEDO プラットフォームロボットPJ





# ソフトウェア特許・ライセンス調査検討委員会

- よく利用されるロボット用ソフトウェアの特許・ライセンスを調査
- リスト化・ガイドライン化
- 開発プロセスへの組み込みを提案
- OSSのより一層の活用とロボット市場拡大の促進



aibo

約500以上のOSSライセンス  
をWebサイト上で明記  
ROS kinetic を利用している  
模様



<http://oss.sony.net/License/ERS-1000/>

# まとめ

- RTミドルウェアの概要
  - 基本概念
  - OSSにまつわる課題
- ROS/ROS2/ROS-industrial
- ロボット新戦略と新規NEDOプロジェクト
  - オープンソース活用によるロボットのプラットフォーム化
  - 調査検討委員会