

2017年7月20日（水）  
東京都立産業技術研究センター  
ロボット用ミドルウェアを活用したソフトウェア開発入門



# ロボティクス・ソフトウェア分野における RTミドルウェアの位置づけと今後

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

# 本日のプログラム

時間	内容
13:00 - 14:00	ロボティクス・ソフトウェア分野におけるRTミドルウェアの位置づけと今後 担当：安藤慶昭（産業技術総合研究所）
14:00 - 15:00	RTミドルウェアの基本
15:00 - 16:00	RTミドルウェアとハードウェア - T型ロボットベース（4輪台車）、センサ
16:00 - 17:00	RTミドルウェアによるロボットシステムの構成

# はじめに

- RTミドルウェアの概要
  - 基本概念
- OpenRTM-aist-1.2の新機能と開発ロードマップ
- ROSとの比較、動向
- プラットフォームロボットプロジェクト
- RTMコミュニティ活動
- まとめ

# RTミドルウェアとは？

# 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

# ロボットミドルウェアについて

- ロボットシステム構築を効率化するための共通機能を提供する**基盤ソフトウェア**
  - 「ロボットOS」と呼ばれることもある
  - インターフェース・プロトコルの共通化、標準化
  - 例として
    - モジュール化・コンポーネント化フレームワークを提供
    - モジュール間の通信をサポート
    - パラメータの設定、配置、起動、モジュールの複合化（結合）機能を提供
    - 抽象化により、OSや言語間連携・相互運用を実現
- 2000年ごろから開発が活発化
  - 世界各国で様々なミドルウェアが開発・公開されている

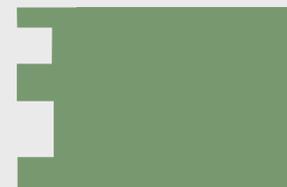
# 従来のシステムでは…



Joystick



Joystick  
software



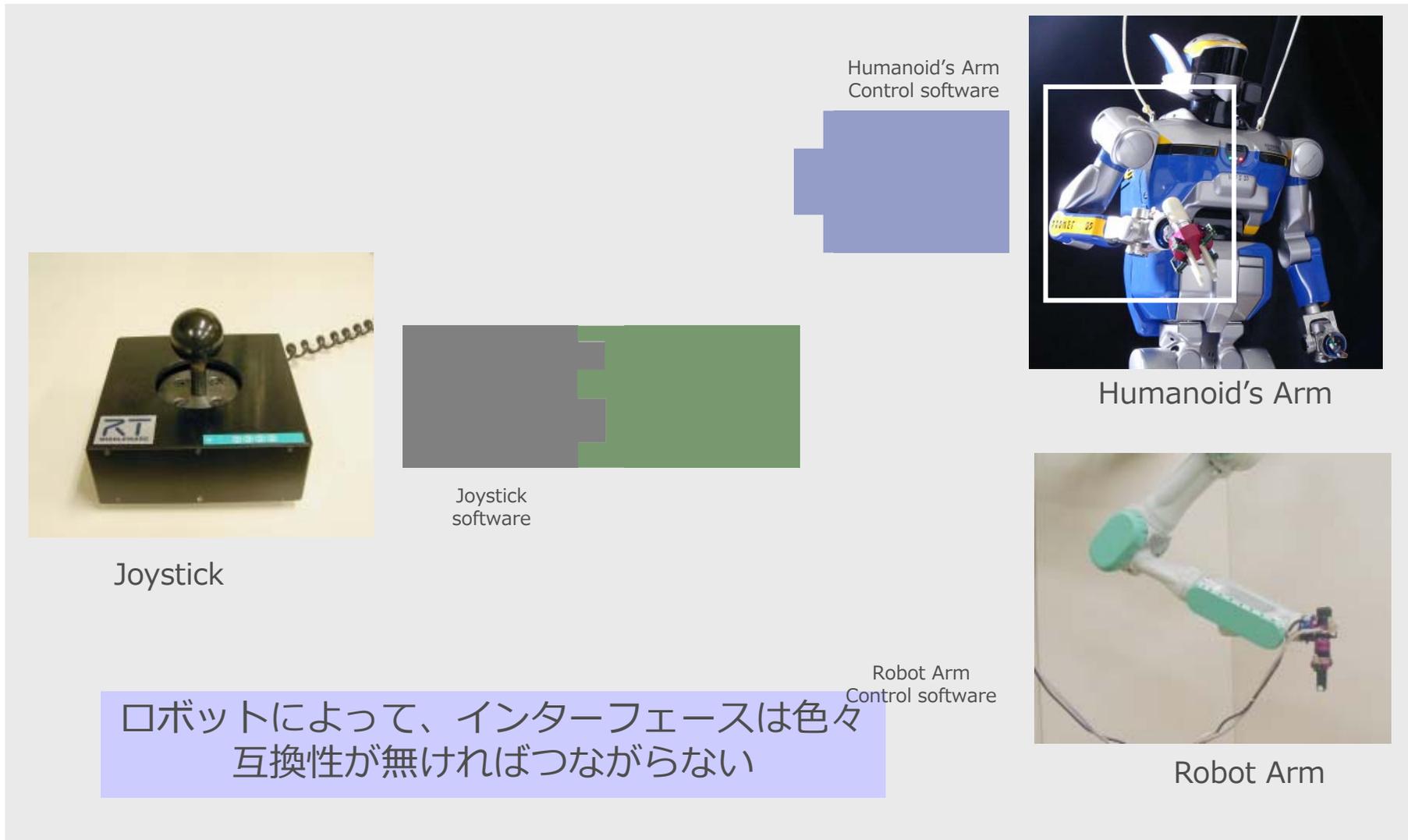
Robot Arm  
Control software



Robot Arm

互換性のあるインターフェース同士は接続可能

# 従来のシステムでは…

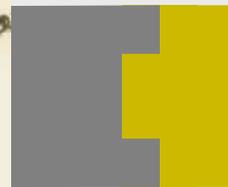


# RTミドルウェアでは…

RTミドルウェアは別々に作られたソフトウェアモジュール同士を繋ぐための共通インターフェースを提供する



Joystick



Joystick software

Arm A  
Control software



Humanoid's Arm

compatible  
arm interfaces



Arm B  
Control software



Robot Arm

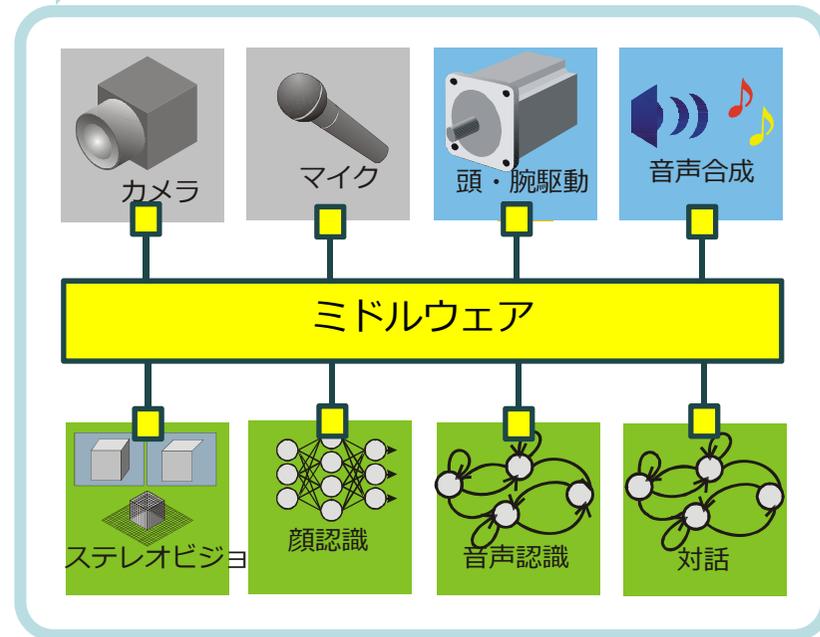
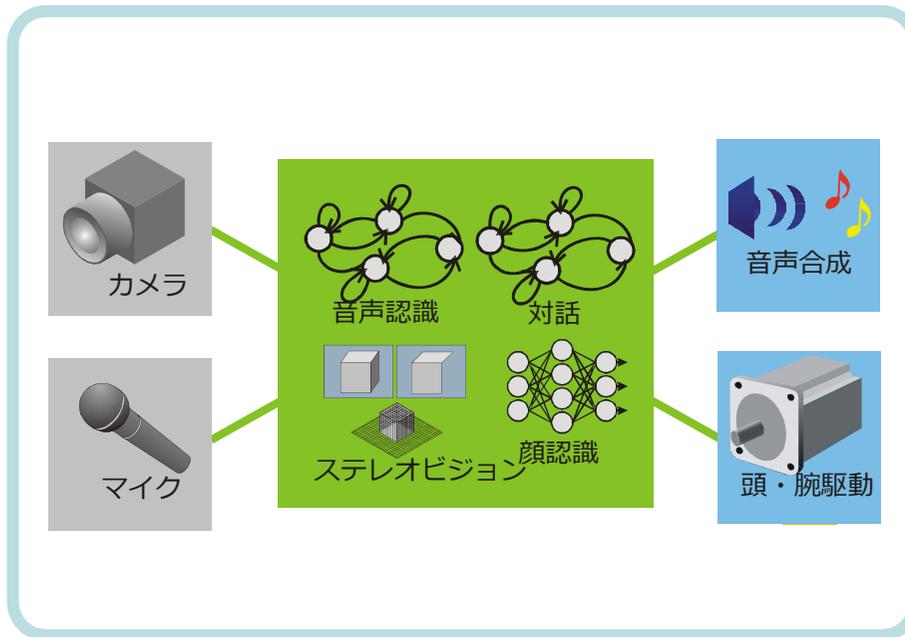
ソフトウェアの再利用性の向上  
RTシステム構築が容易になる

# 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

# モジュール化のメリット

- 再利用性の向上
  - 同じコンポーネントをいろいろなシステムに使いまわせる
- 選択肢の多様化
  - 同じ機能を持つ複数のモジュールを試すことができる
- 柔軟性の向上
  - モジュール接続構成かえるだけで様々なシステムを構築できる
- 信頼性の向上
  - モジュール単位でテスト可能なため信頼性が向上する
- 堅牢性の向上
  - システムがモジュールで分割されているので、一つの問題が全体に波及しにくい

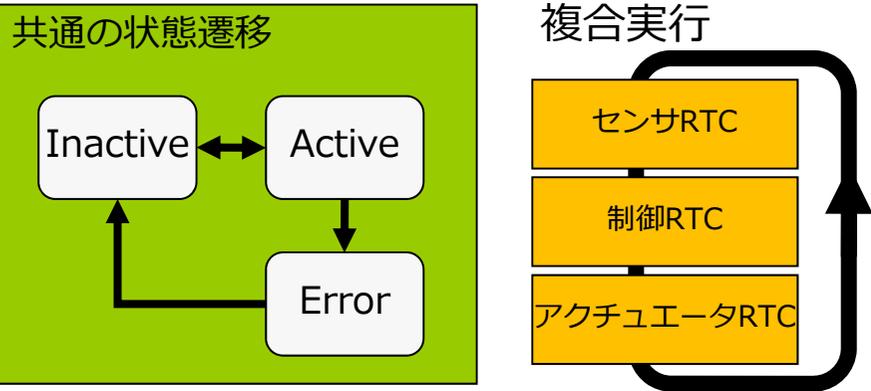
# RTコンポーネント化のメリット

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

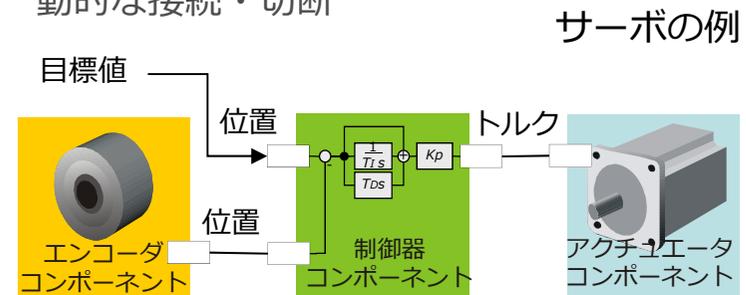
## アクティビティ・実行コンテキスト



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

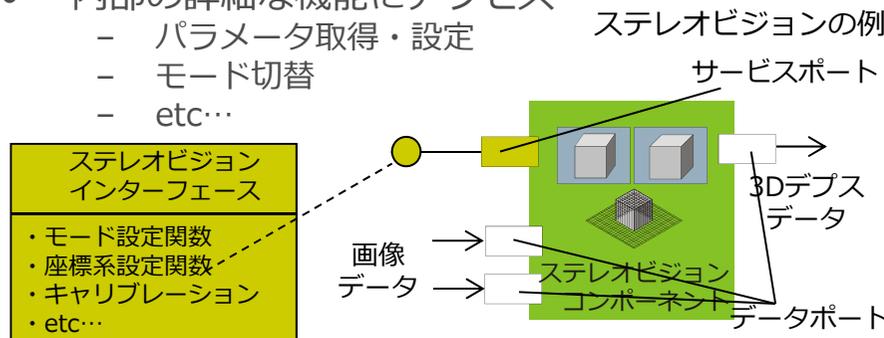
- データ指向ポート
- 連続的なデータの送受信
- 動的な接続・切断



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...

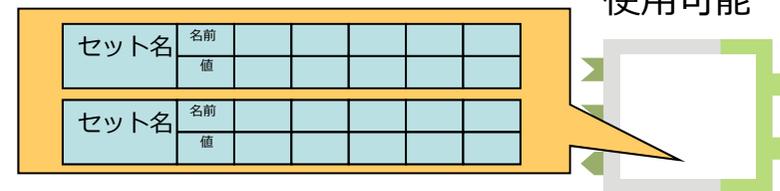


サービス指向相互作用機能

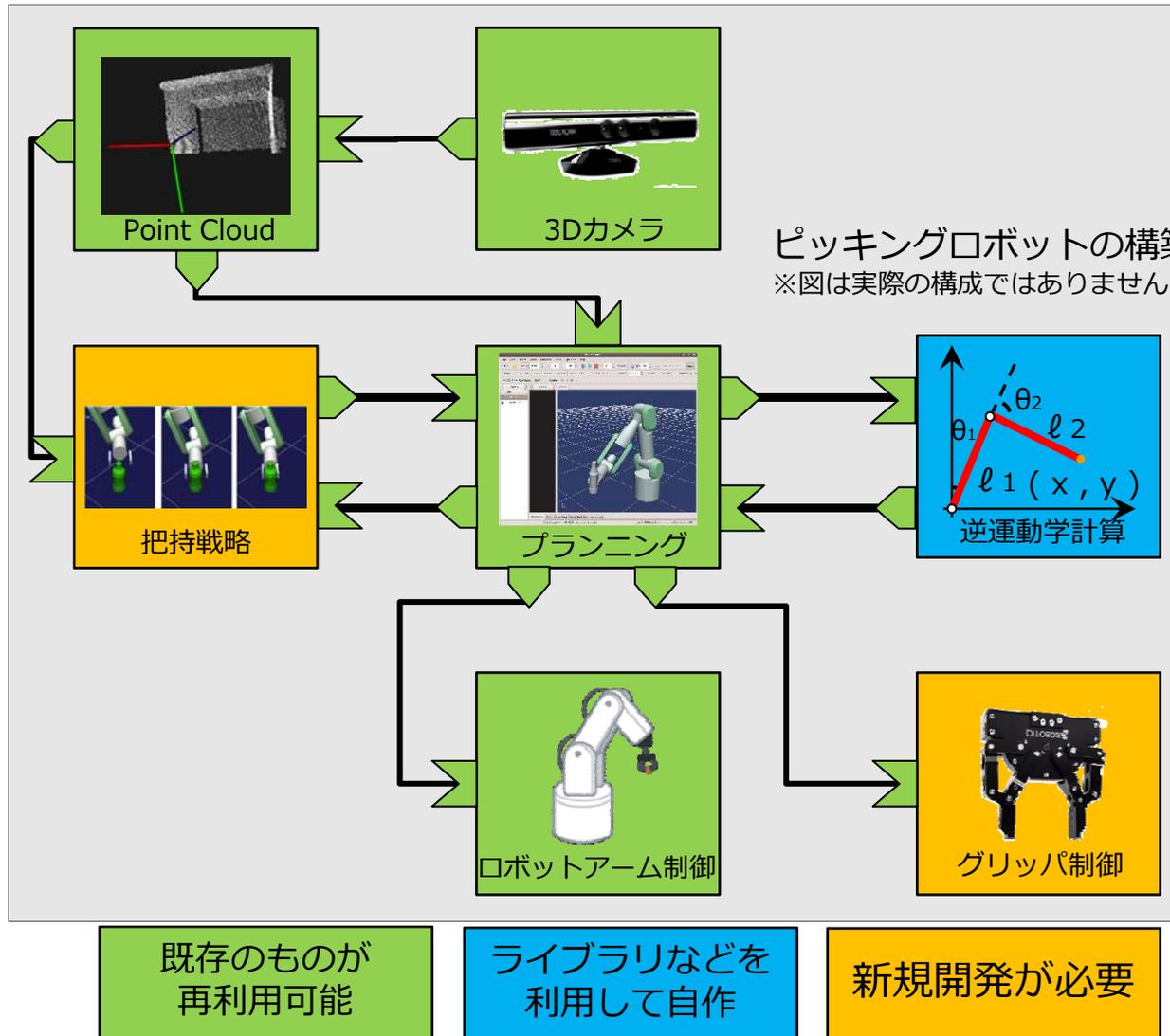
## コンフィギュレーション

- 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

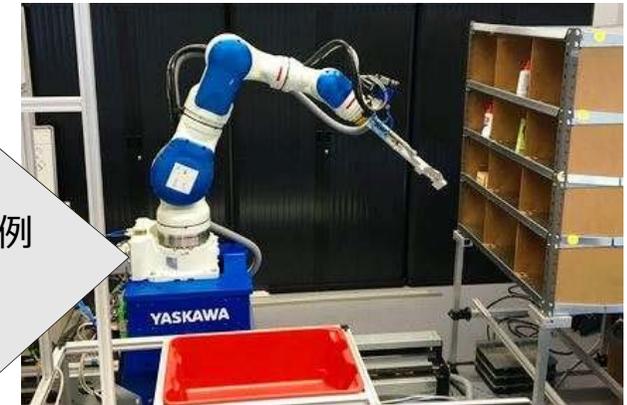
複数のセットを  
動作時に  
切り替えて  
使用可能



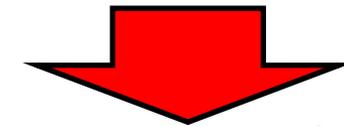
# ミドルウェアを利用した開発の利点



ピッキングロボットの構築例  
※図は実際の構成ではありません。

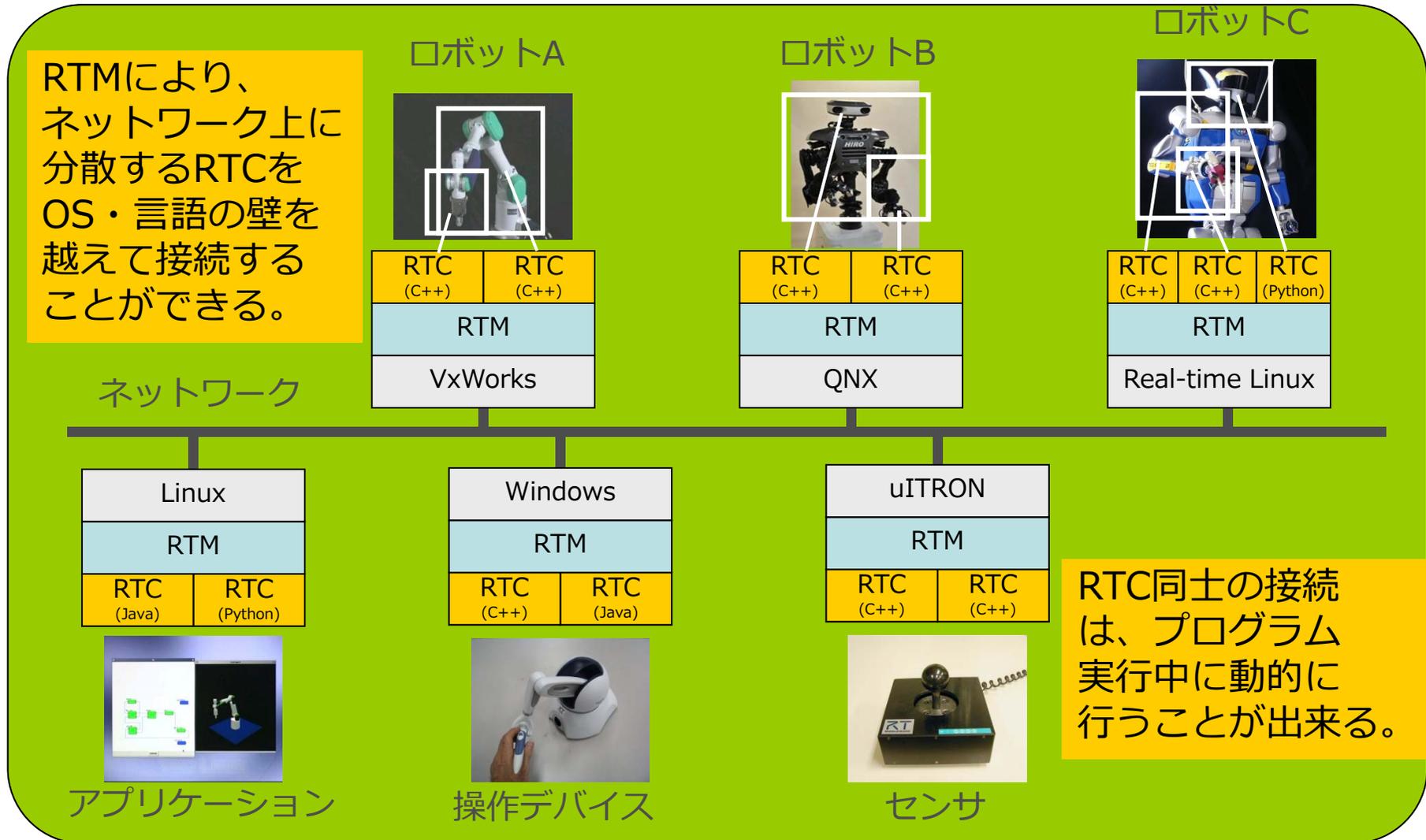


ミドルウェアを利用すると、**既存のモジュール**が利用できる



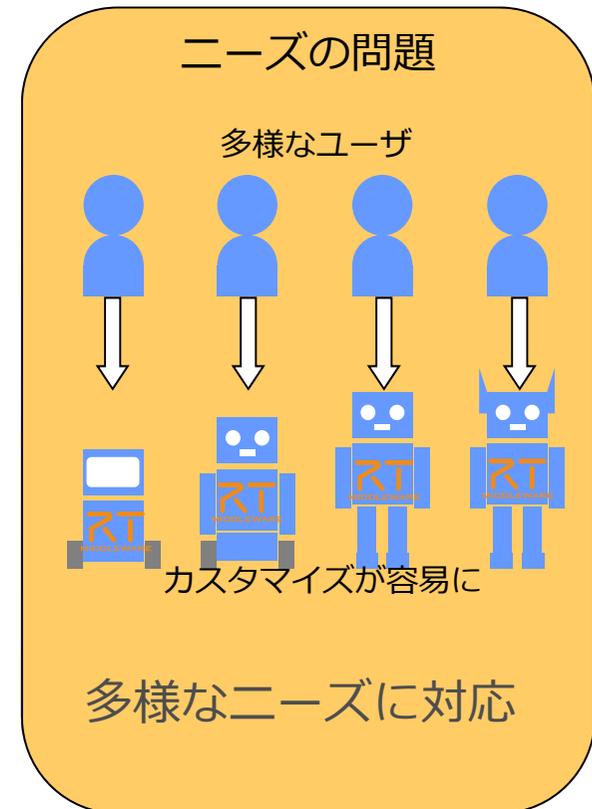
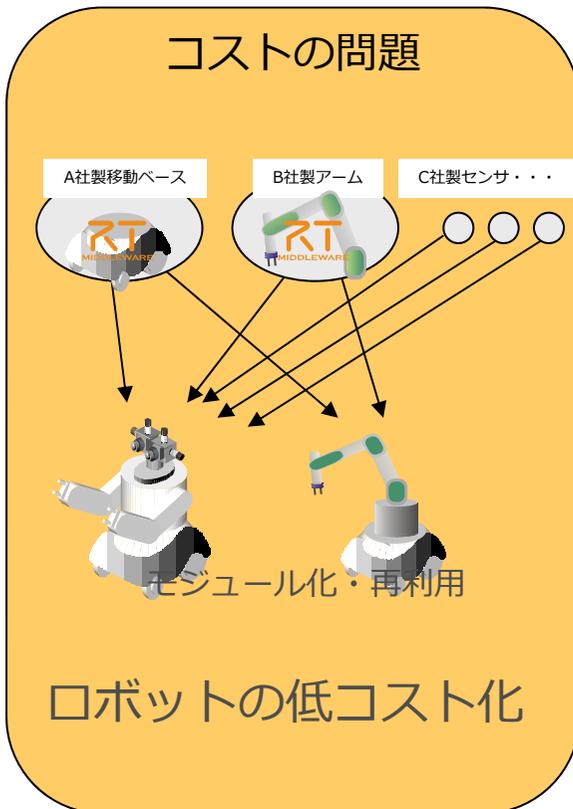
開発するとき**新規に作らなければならない部分**は少なくて済む

# 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

# 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



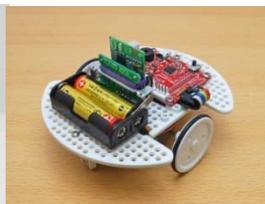
HIRO, NEXTAGE open: Kawada Robotics



RAPUDA : Life Robotics



ビュートローバーRTC/RTC-BT(VSTONE)



OROCHI (アールティ)



新日本電工他: Mobile SEM

# RTミドルウェアは国際標準

## OMG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

- 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

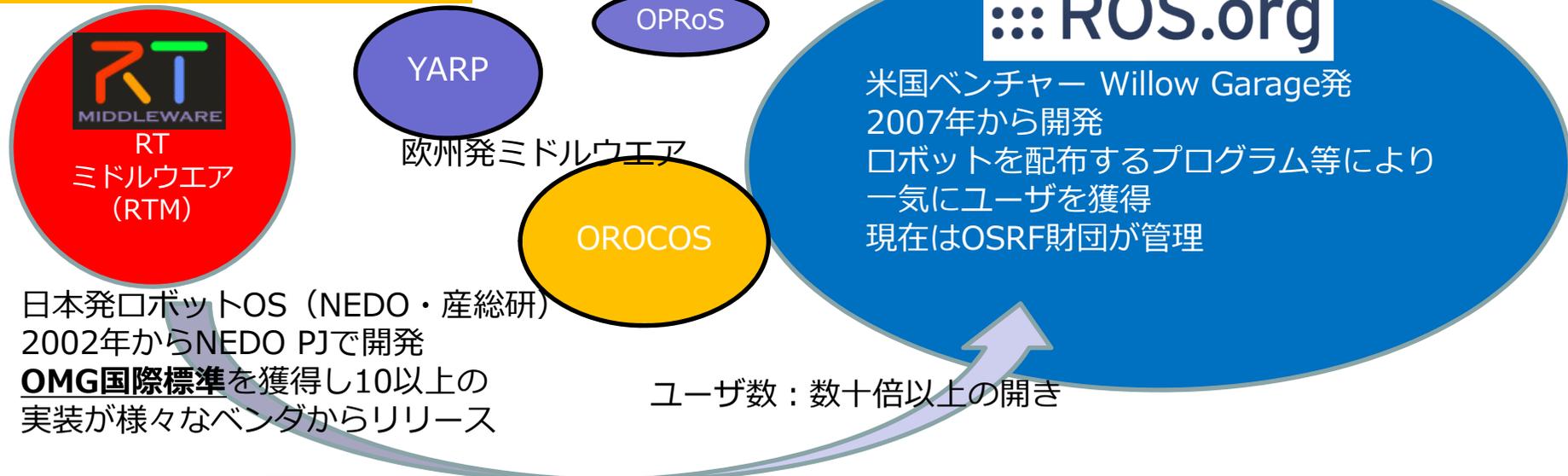
名称	ベンダ	特徴	互換性
OpenRTM-aist	産総研	NEDO PJで開発。参照実装。	---
HRTM	ホンダ	アシモはHRTMへ移行中	◎
OpenRTM.NET	セック	.NET(C#,VB,C++/CLI, F#, etc..)	◎
RTM on Android	セック	Android版RTミドルウェア	◎
RTC-Lite	産総研	PIC, dsPIC上の実装	○
Mini/MicorRTC	SEC	NEDOオープンイノベーションPJで開発	○
RTMSafety	SEC/AIST	NEDO知能化PJで開発・機能安全認証取得	○
RTC CANOpen	SIT, CiA	CAN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

# ROSとの比較、動向

# ロボットOS：現状と課題

## ロボットOS勢力図



## 課題



- OROSはデファクト標準であり、OSRFにより仕様を変えられてしまう恐れあり。
  - ⇒ バージョンが上がるごとに、ユーザが振り回される
  - ⇒ ROSは技術的に古く、次期バージョンアップで大幅に変わる可能性あり。→ROS2へ
- 国内ユーザと海外ユーザの利用OSの差 (国内：Windows, ITRON, 海外：Linux)
  - ⇒ Linuxのみ対応するROSと、様々なOSに対応するRTMの利用比率は半々程度
- 国内でのソフトウェアの再利用が進んでいない。
  - ⇒ オープンソース文化の欠如。
  - ⇒ NEDO知能化PJで蓄積したソフトウェア資産も再利用が進んでいない。

# 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以外にもサポート
- 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
- 英語の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予定。

# ROS2

- 最新版：2016年12月 Beta版リリース
- 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)

# ロボット新戦略(2015年2月)

- 日本経済再生本部決定文書
  - ロボット革命実現会議の取りまとめ

pp.27-32

## 第2部 アクションプランー 五カ年計画

### 第1章 分野横断的事項

#### 第3節 ロボット国際標準化への対応

##### 第1項 ハードウェア/ソフトウェアのモジュール化を見据えた共通基盤 (ミドルウェア・ロボット OS)

(中略) ロボットを構成するハードウェア、ソフトウェアの機能要素をモジュール化・共通化することで、(中略) ロボットを低価格で構成することが可能となる。また、共通のソフトウェアプラットフォームを利用することで、(中略) ロボットインテグレータは対象毎の機能の実現に注力することができる。

#### (3) 2020年に目指すべき姿

ロボットによる産業革命を日本主導で実現するためにも、(中略) ロボット制御の基幹部分であるロボット OS・ミドルウェアについても積極的な取組を進めていくことが重要である。(中略) 様々な国際標準を日本主導で策定することで、(中略) 日本のロボット産業が世界をリードすることが可能となる。

(中略) そのため、新たに設置される「ロボット革命イニシアティブ協議会」(第1項参照)が中心となって、必要な情報収集及び国際発信を行い、国際標準化の流れを主導していく。

## ロボット新戦略

Japan's Robot Strategy

ービジョン・戦略・アクションプランー

日本経済再生本部  
2015年2月10日

## ロボット新戦略の重要性

関係各省庁も議論に参加し、内容の詳細に関与し合意したロボット戦略

→今後の施策に大きな影響

# 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

製造産業局 ロボット政策室  
03-3501-1049

平成29年度予算案額 17.5億円 (15.0億円)

**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

交付 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

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	システムインテグレーション費用
ソフトウェア		ソフトウェア
周辺装置		周辺装置
ロボット本体		ロボット本体

現状 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  
5月1日に公募開始

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

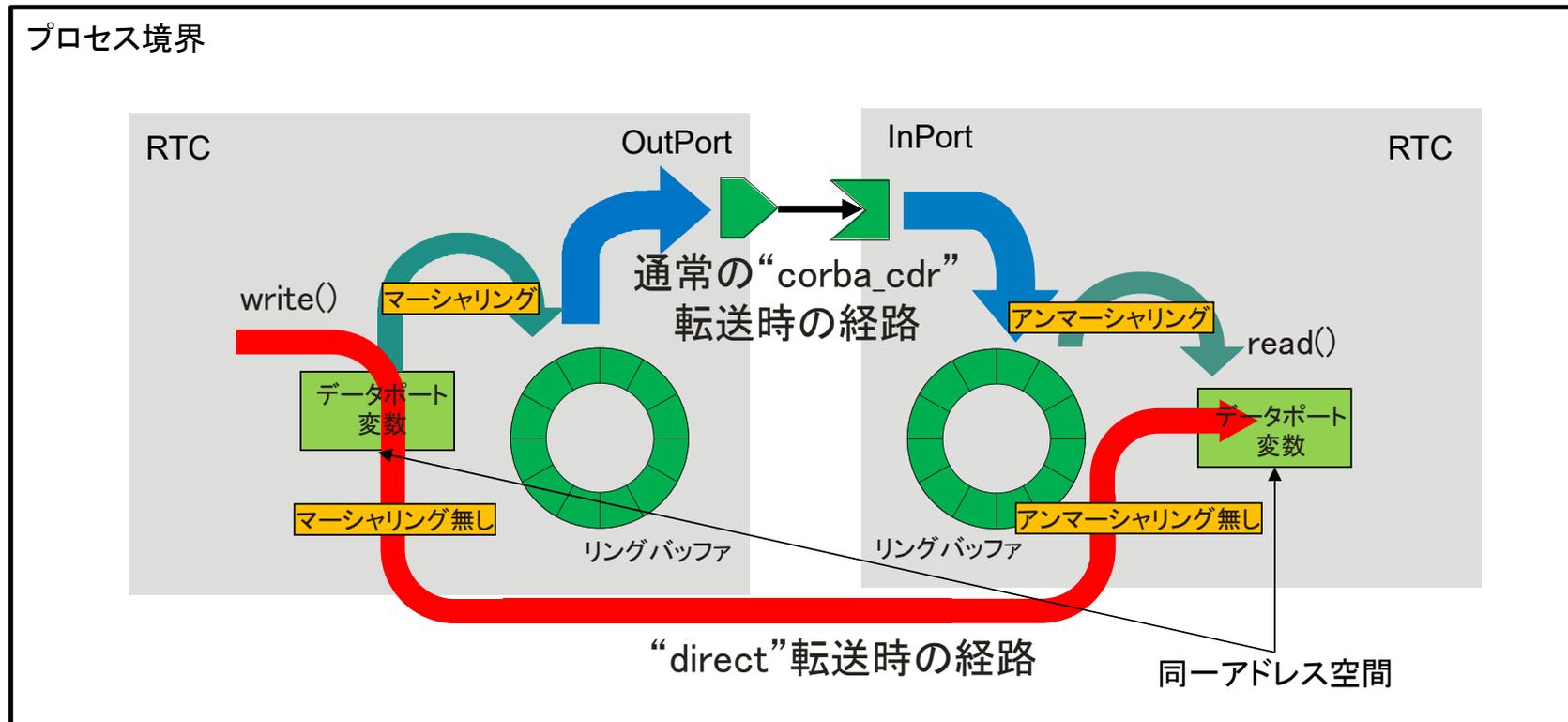
1

### 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。」

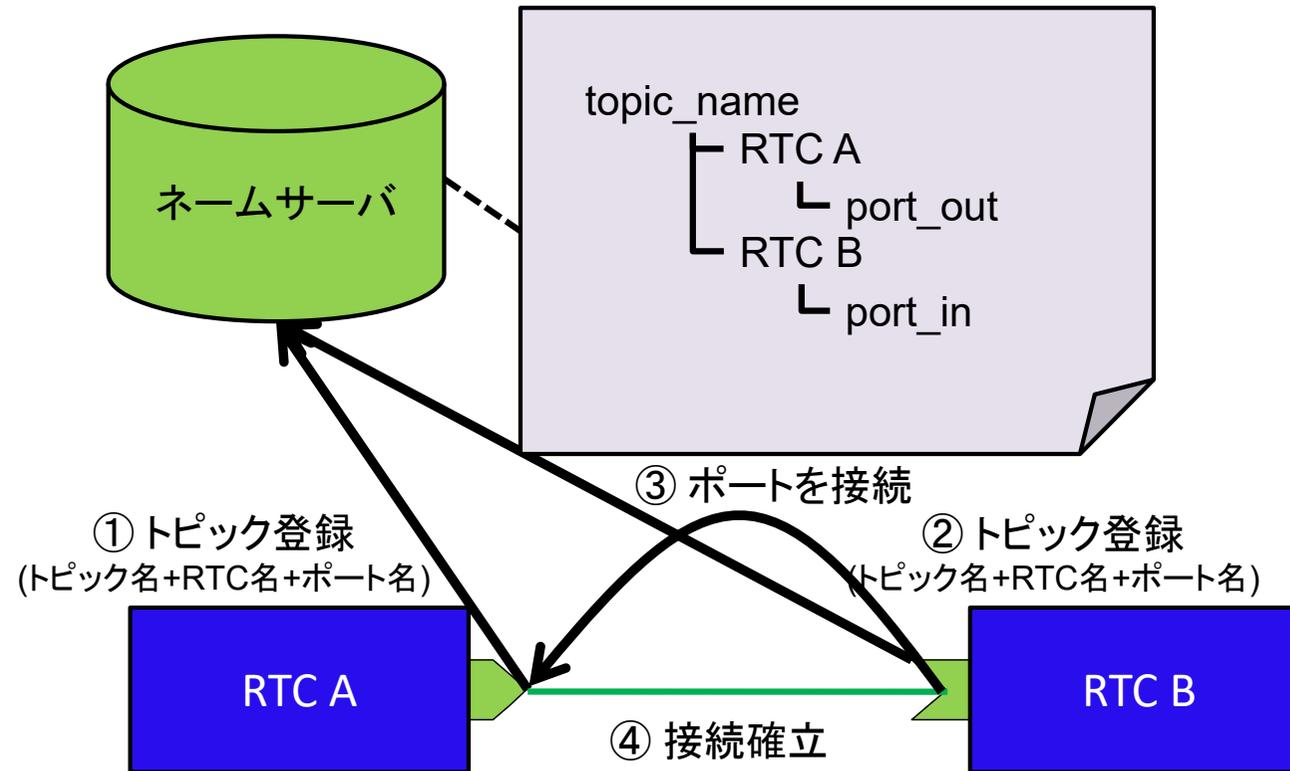
# OpenRTM-aist-1.2の新機能 と今後の開発ロードマップ

# ダイレクト接続



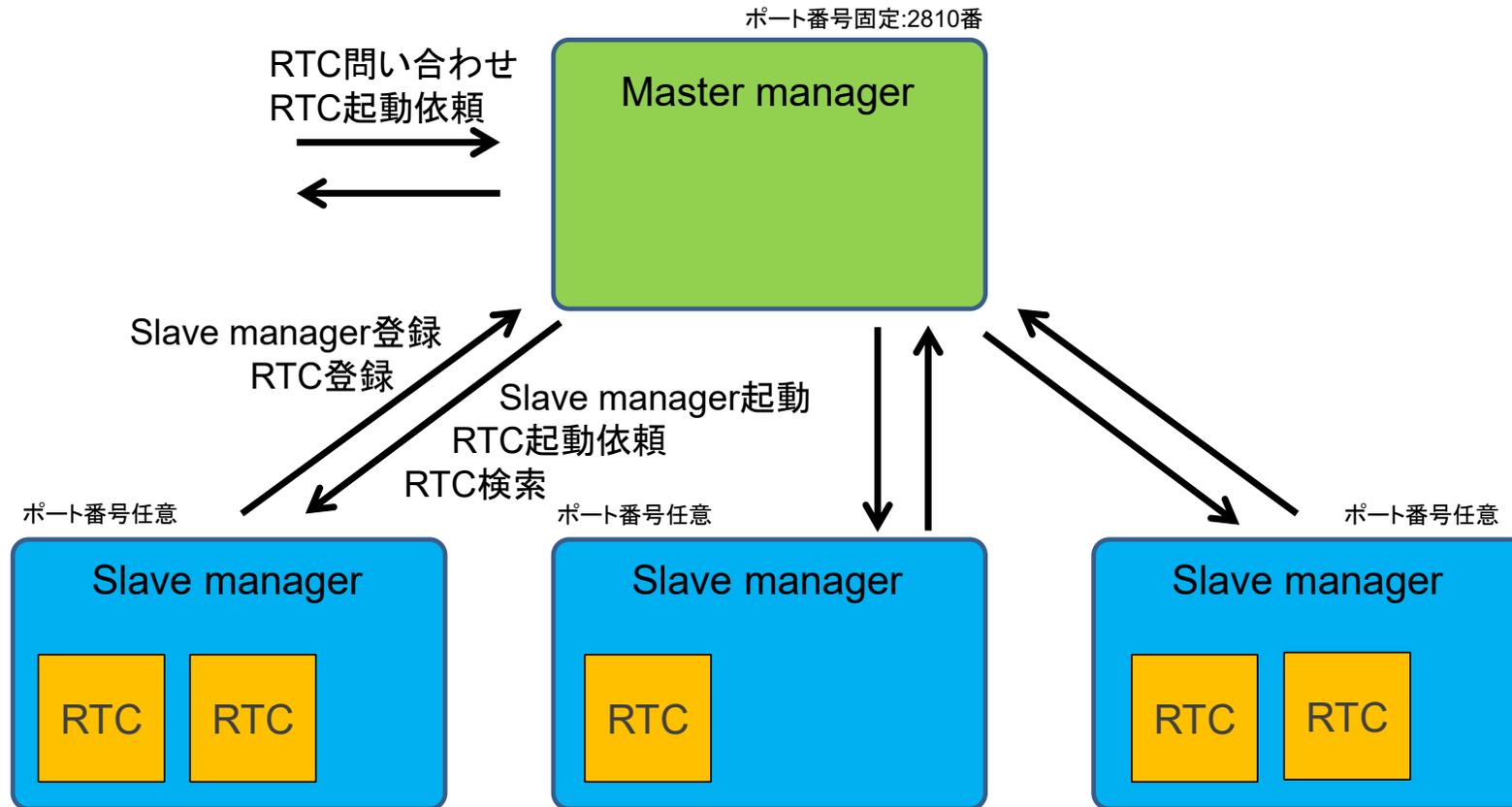
**Direct接続**：バッファ、マーシャリングをバイパスして直接変数領域でデータを受け渡し（変数間コピーなので高速）

# トピック接続



**トピック接続**：相手のポートを特定せずに、同一のトピック（ラベル）を持つポート間で自動接続を行う機能。

# マネージャ (rtcd) 機能の拡張



コンポーネントの起動~終了（ライフサイクル）をリモートからすべて制御可能に。多数のノードを統一的に管理可能となり、運用時の効率向上が図れる。

# その他の変更

- Windows
  - VS2017対応
  - VCバージョンチェンジャーツール提供
  - 32bit、64bit同時インストール可能
  - 同梱OpenCV 3.2へ変更
  - 同梱JRE OpenJDK8へ変更
  - Cmakeによるmsiパッケージ作成が可能
- Linux
  - OpenRTM-aist-Java, OpenRTPパッケージ提供
  - インストールスクリプト1本化 (C++, Python, Java, ツール)
  - Cmakeによる deb/rpmパッケージ作成が可能
  - Ubuntu (12.04, 14.04, 16.04, 16.10, 17.04)
  - Debian (7.0, 8.0)
  - Fedora (23,24,25)

# 2.0に向けて

- 新標準FSM4RTCの正式サポート
  - 状態遷移型RTCの枠組み
  - 5月12日 2A2J10 (RT ミドルウェアとオープンシステム) にて発表
- ROSやその他のミドルウェアとの連携機能
  - 要望多数
  - データポート相互接続
  - 運用時互換性等
- 開発効率向上のための機能
  - パッケージ化
  - 自動ビルド・CI
- 運用効率向上のための機能
  - ロギング機能、構成管理機能
  - RTC合成、組み込み機能
- サードパーティーRTC・ツール取り込み
- ROSノードの取り込み



Version 2.0  
2017年度末

Version 2.1  
2018年度中

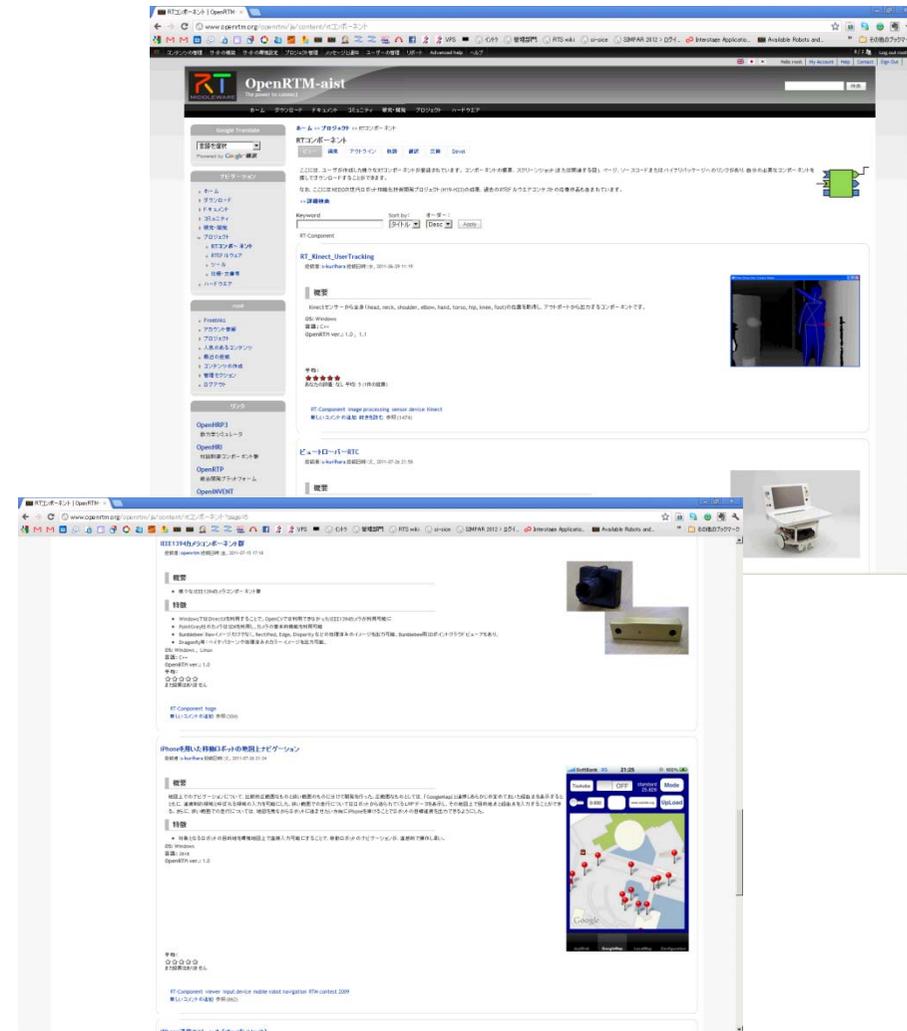
Version 2.2  
2018~2019年度

# RTミドルウェアと コミュニティ

# プロジェクトページ

- ユーザが自分の作品を登録
- 他のユーザの作ったRTCを探ることができる

タイプ	登録数
RTコンポーネント群	287
RTミドルウェア	14
ツール	19
仕様・文書	4
ハードウェア	28



# サマーキャンプ

- 毎年夏に1週間開催
- 今年：7月31日～8月4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



# 提言

- 自前主義はやめよう！！
  - 書きたてのコードより、いろいろな人に何万回も実行されたコードのほうが動くコードである！！
  - 自分にとって本質的でない部分は任せて、本当にやりたい部分・やるべき部分のコードを書こう！！
  - 誰かがリリースしたプログラムは一度は動いたことがあるプログラムである！！
  - 人のコードを読むのが面倒だからと捨ててしまうのはもったいない！！
- オープンソースにコミットしよう！！
  - 臆せずMLやフォーラムで質問しよう！！
  - どんなに初歩的な質問でも他の人にとっては価値ある情報である。
  - 要望を積極的にあげよう！！
  - できればデバッグしてパッチを送ろう！

# まとめ

- RTミドルウェアの概要
  - 基本概念
- OpenRTM-aist-1.2の新機能
- 2.0以降の開発ロードマップ
- ROSとの比較、動向
- ロボット新戦略と新規NEDOプロジェクト
- RTMコミュニティー活動