次世代ロボット フレームワーク ROS2の紹介

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
Geoffrey BIGGS

Part 1

ロボット
ソフトウェア

ロボットソフトウェアの特質

- 複数の分野の混合
  - 制御理論、信号処理、自動計画、AI、…
- リアルタイムと非リアルタイムの混合
- 大量のデータ処理
- CPU、メモリは甚だしく利用も、
  小さなマイコン利用
- ハードウェアとソフトウェア
  アーキテクチャーの複雑さ
- 現実世界との複雑なインタラクション
- …

ロボットソフトウェアフレームワークの特質

- ソフトウェアコンポーネント
- 分散可能
- リアルタイム可能
- 大量データ処理可能

- オープンソース

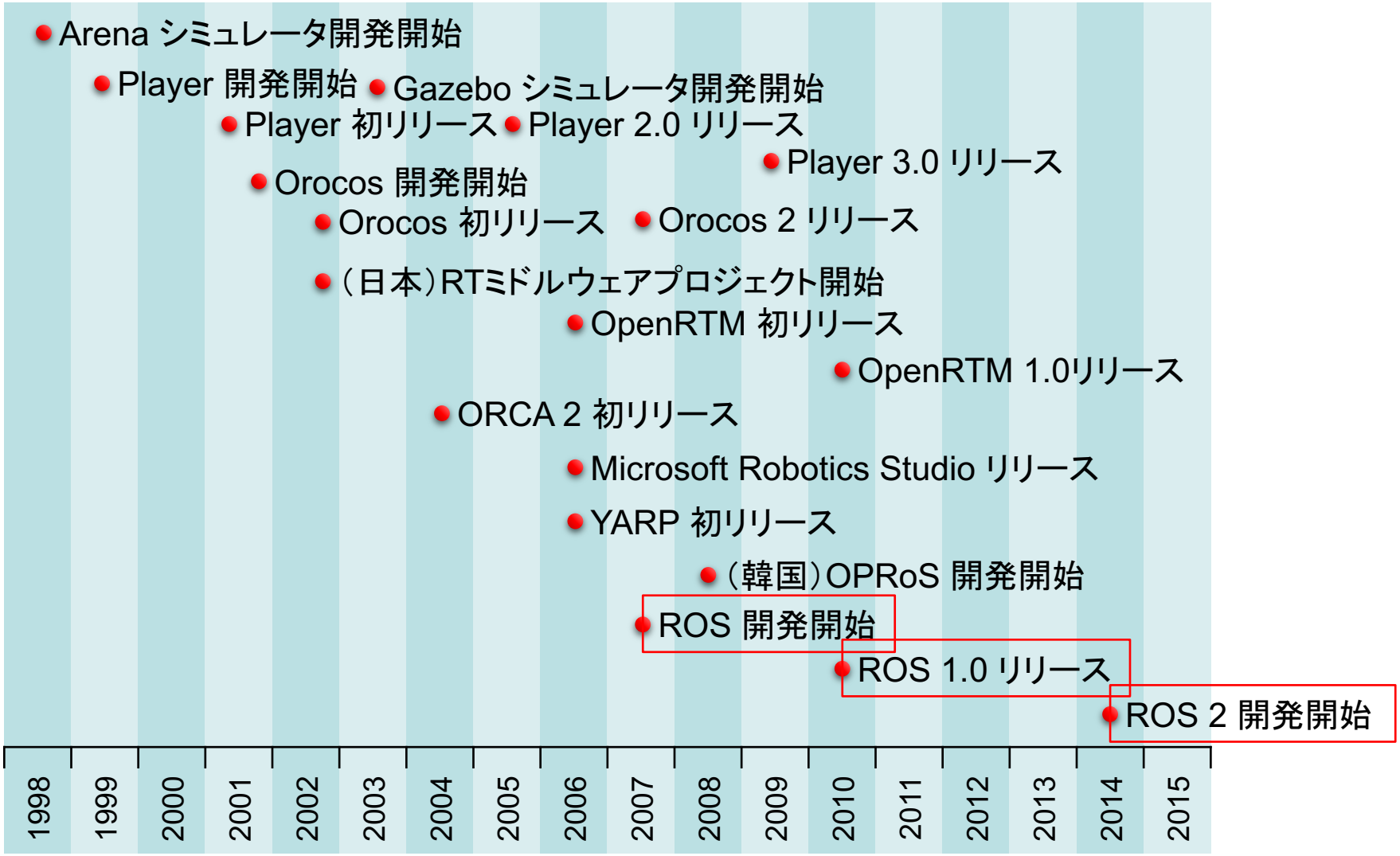
なぜソフトウェアコンポーネント？

- 分野を超えるシステム
  - 一人で作れない
  - 世界じゅうに分布された開発者からのソフトウェアのインテグレーション
- ロボットの作成者は数年しかいない
  - 大学では「PhD Problem」
  - 企業では「Promotion Problem」
  - だれでも簡単に利用できるように
- システムレベルで考えたい

➡ APIとライブラリより<u>コンポーネント</u>

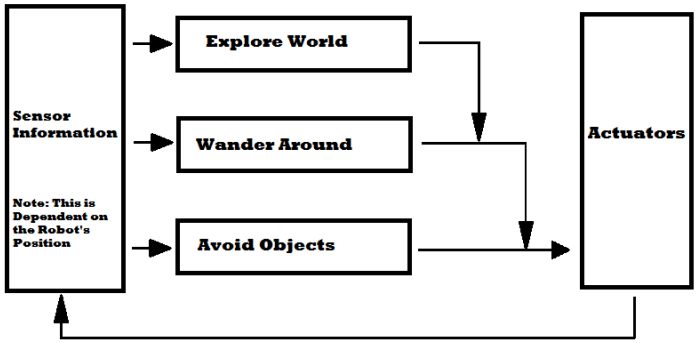
- 再利用レベルが違う
- 「Plug-and-play」

ロボットソフトウェアの歴史



20世紀

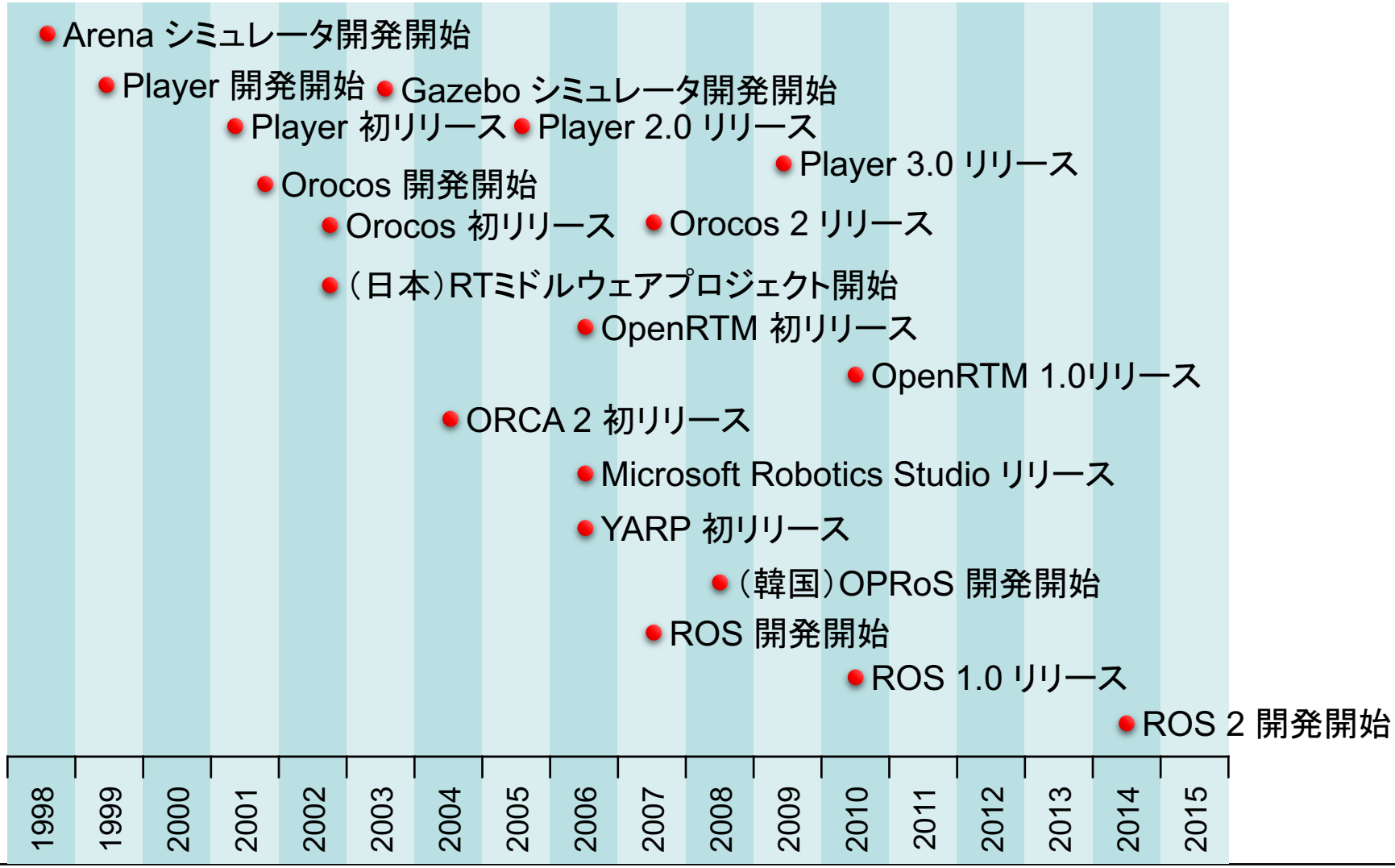
- ソフトウェアは完全にカスタム
  - 作った学生が卒業したら、おしまい
- プロプライエタリ・ソフトウェア
  - ロボットやアクチュエーターやセンサーとのセット
  - ハードウェア（メーカー）によってAPIの変更
- 特定コンセプトに合うソフトウェア
  - そのソフトウェアの方法にしか使えない
  - main()関数等利用不可能
  - 自由なし→研究でもプロトタイプ作成でも利用難い



Playerの登場

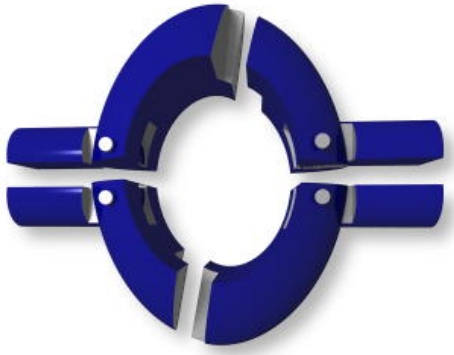
- 世界初のオープンソースロボットソフトウェアフレームワーク
  - 一般的に応用可能：どんなロボットでも利用可能
  - オープンだから研究ニーズに合わせて編集可能
- APIはUNIXモデルに
  - 使いやすい
  - 柔軟
  - 「これがロボットの正しい開発方法だ」より「好きな方法で開発てもいい」

フレームワークブームとコンポーネント化



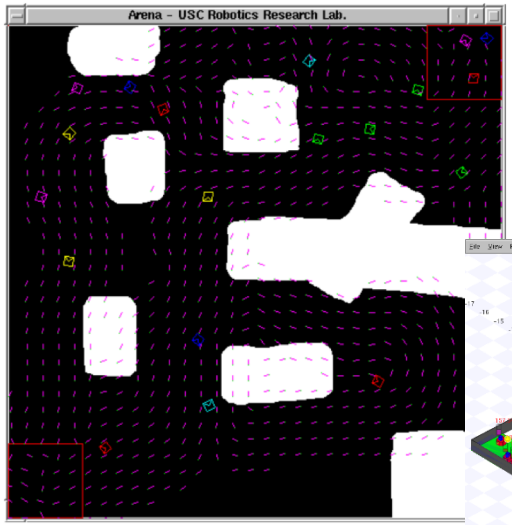
現在

- フレームワークの数が減った
  - YARP
  - Orocos
  - OpenRTM
  - ROS
- それぞれの特徴あるけど、ROSのみはユーザ数とソフトウェア数が多い

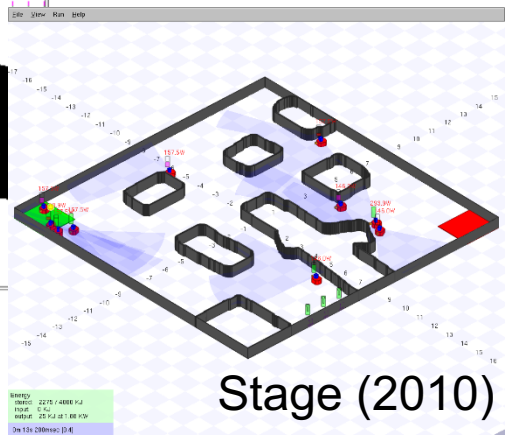


ツールも進んだ

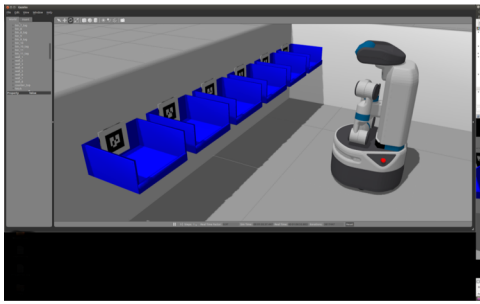
- 20世紀：ほぼなし
- オープンソースロボットシミュレータが最初
  - Arena → Stage + Gazebo
- ROSのツールで、ツールの大事さが明確
  - ROSは技術よりツールのおかげで成功
- 現在の傾向：モデルベースツール
  - ソフトウェア・コンポーネントとフィット



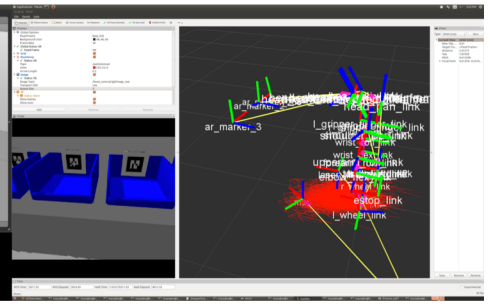
Arena (2000)



Stage (2010)



Gazebo



rviz

Part 2

ROS

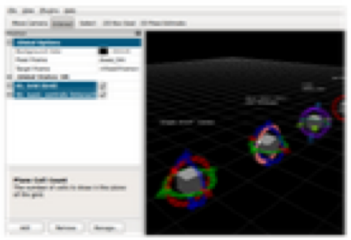
ROS

- Robot Operating System
- 2007年から
- フレームワークだけではなくて、<u>プラットフォーム全体</u>

Plumbing

コンポーネント
間の通信

+



Tools

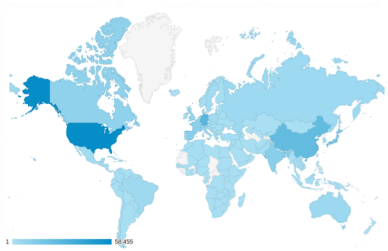
ツール

+

Capabilities

頻繁に利用される
ロボットの機能

+



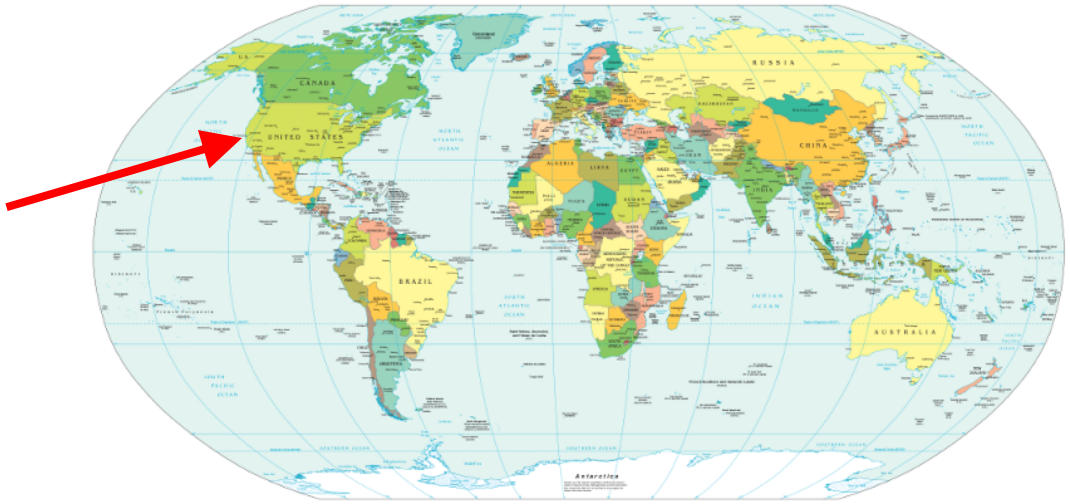
Ecosystem

エコシステムと
コミュニティー

Open Source Robotics Foundation

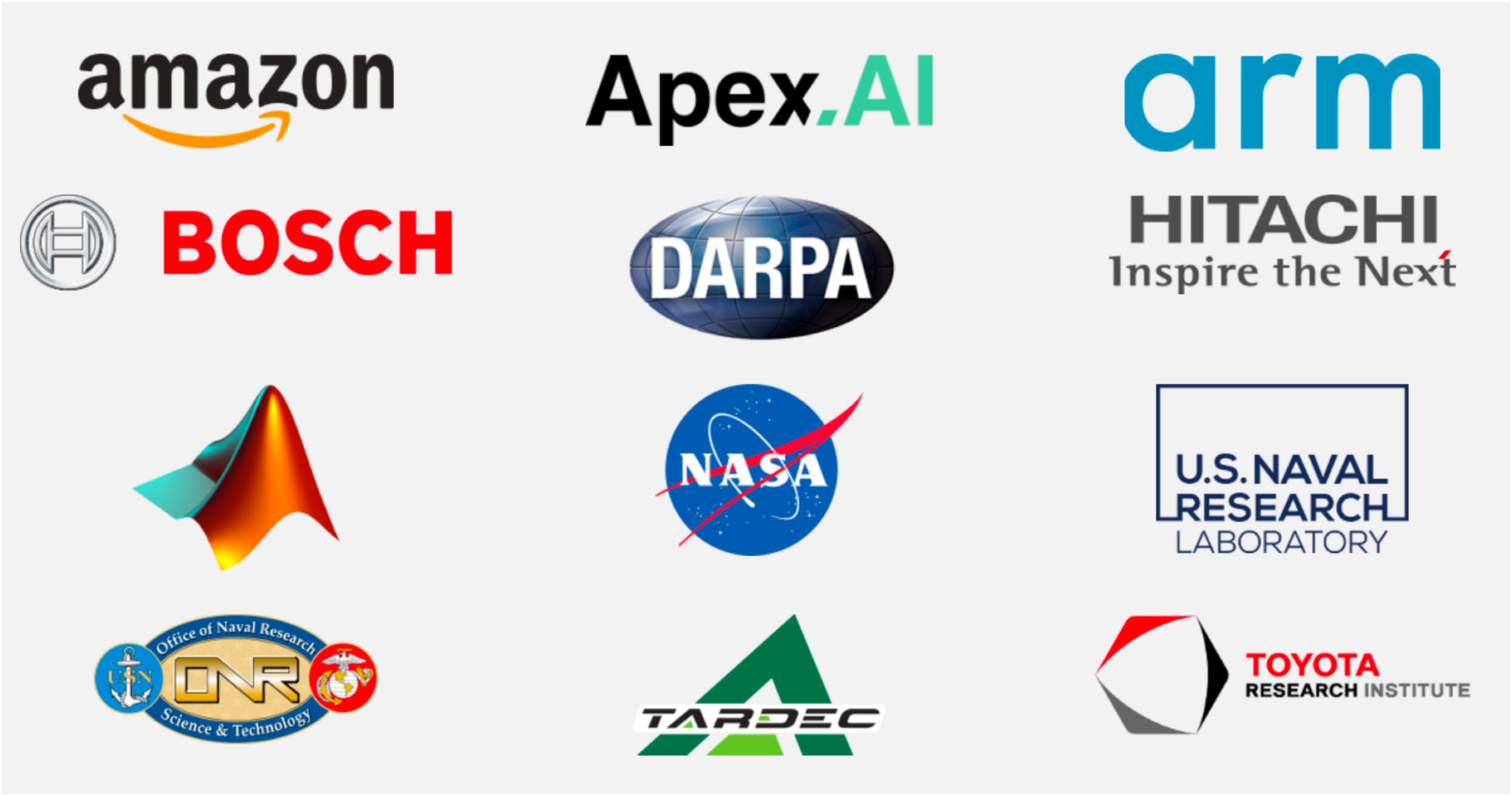
ミッションステートメント:

“...to support the development, distribution, and adoption of open source software for use in robotics research, education, and product development.”



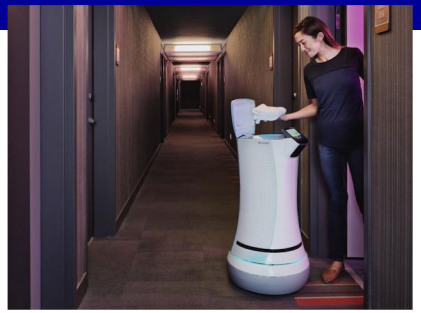
http://osrfoundation.org

OSRFのスポンサー


amazon
Apex.AI
arm
BOSCH
DARPA
HITACHI
Inspire the Next
NASA
U.S. NAVAL RESEARCH LABORATORY
Office of Naval Research
ONR
Science & Technology
TARDEC
TOYOTA
RESEARCH INSTITUTE

ロボット機能の例

一般に応用可能な
2D ナビゲーション
システム



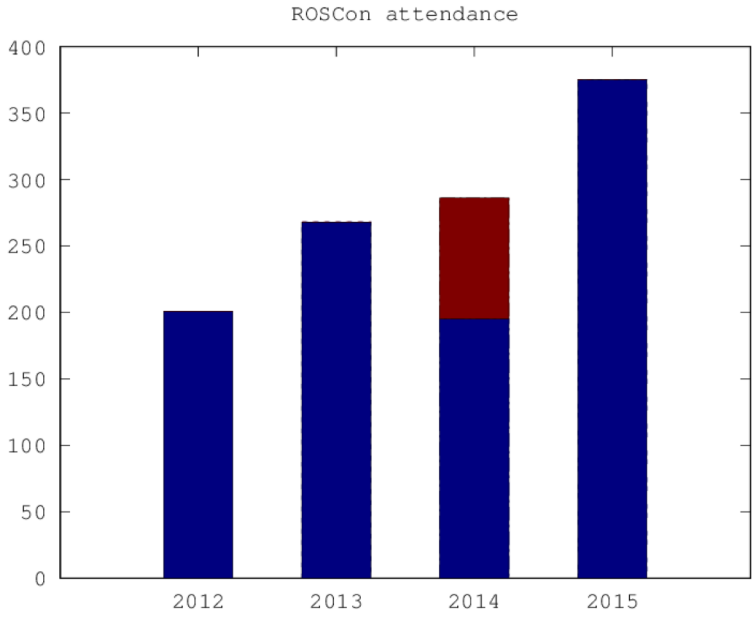




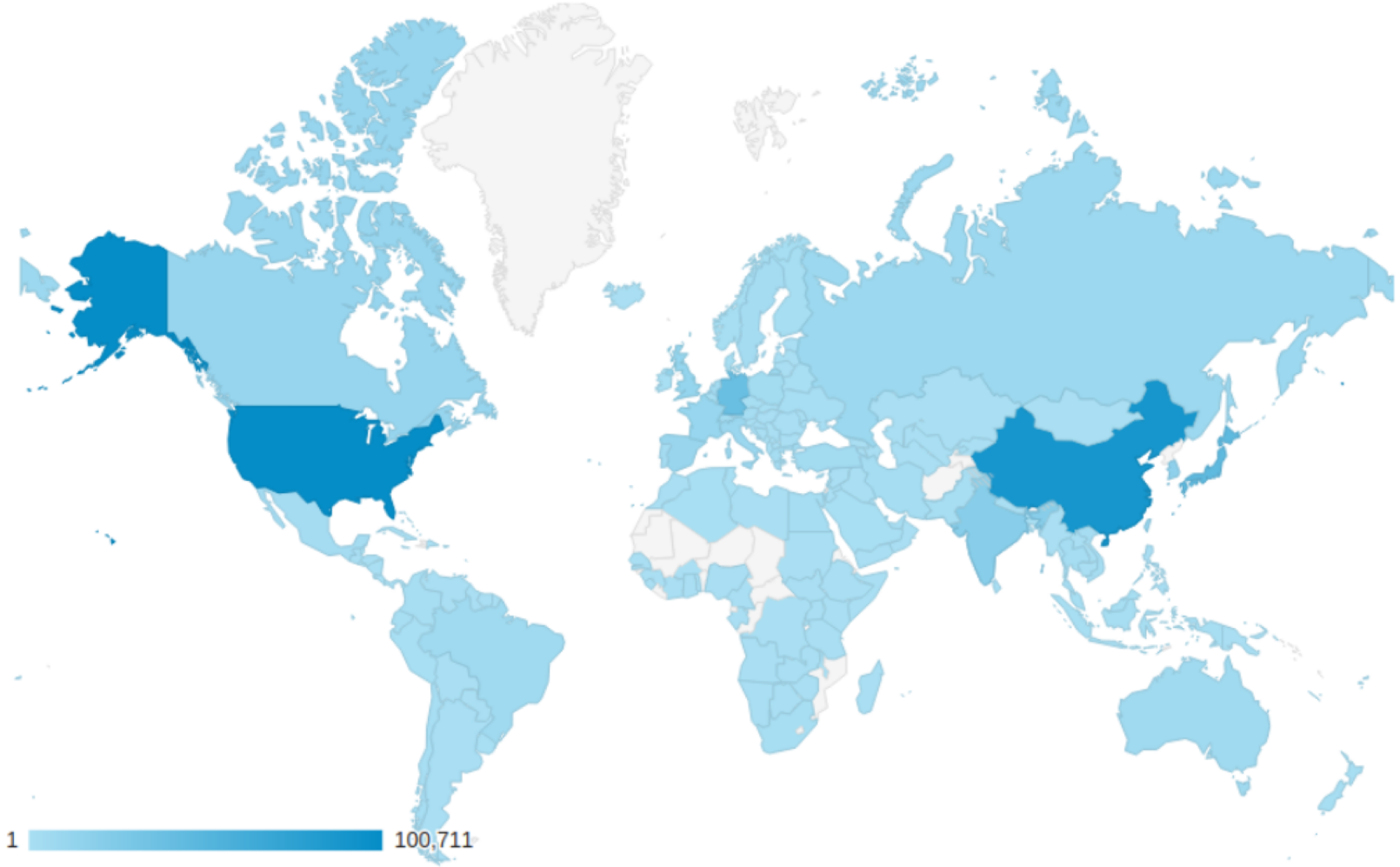



http://wiki.ros.org/navigation

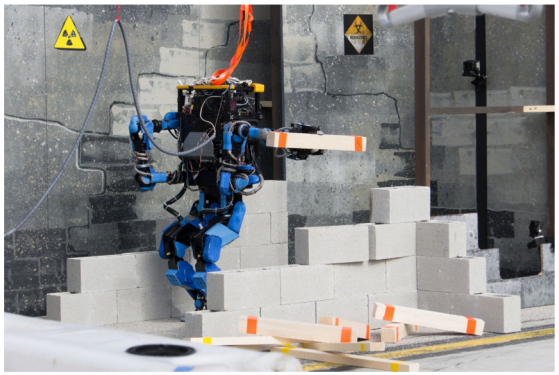
コミュニティーの例: ROSCon

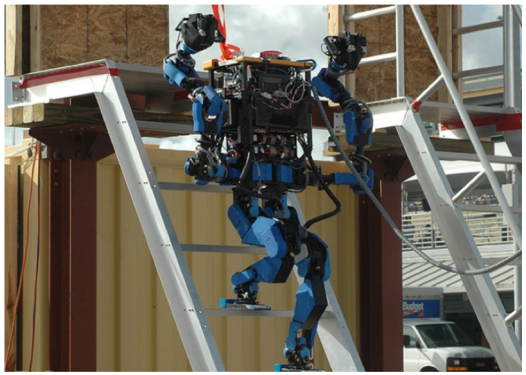


だれがROSを使っている
（2017年版）



DARPA Robotics Challenge: 2012–2015





DRC決勝の23チームの中で:

18 チームが

ROS

を利用した

BMW: CES 2015 自動運転デモ

ROS

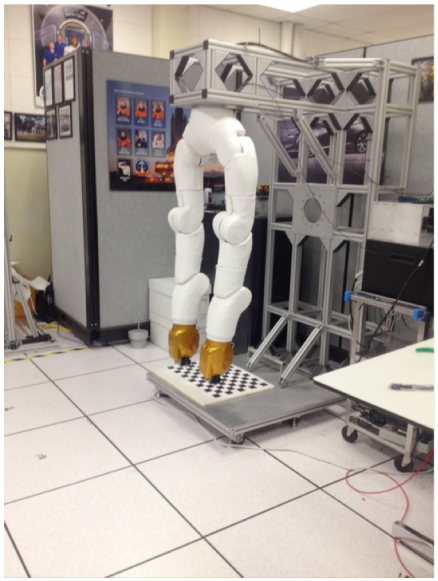
Yujin Robot: 病院で薬等の配達

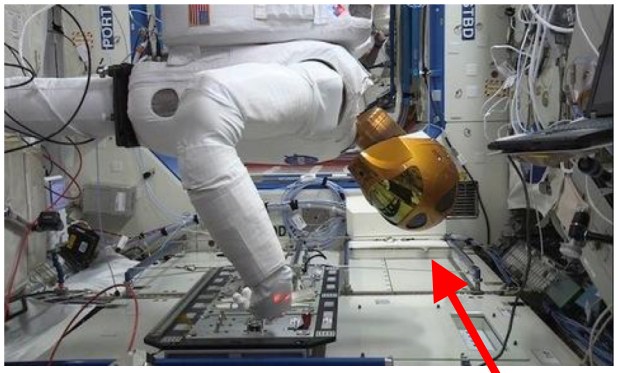
ROS

Rethink Robotics: Baxter と Sawyer

ROS

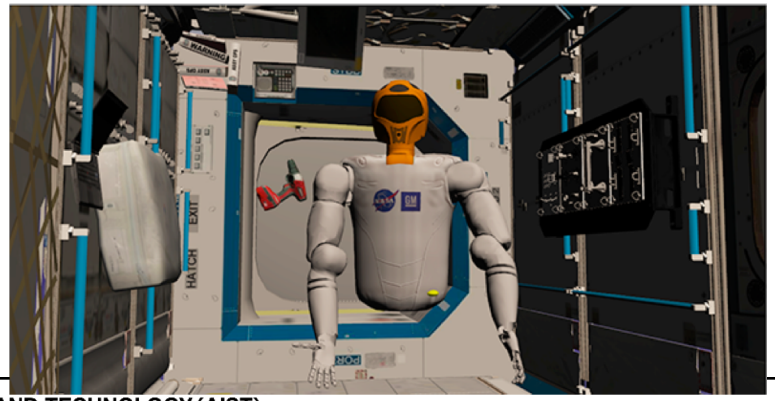
NASA: Robonaut 2
（国際宇宙ステーションで）





GAZEBO

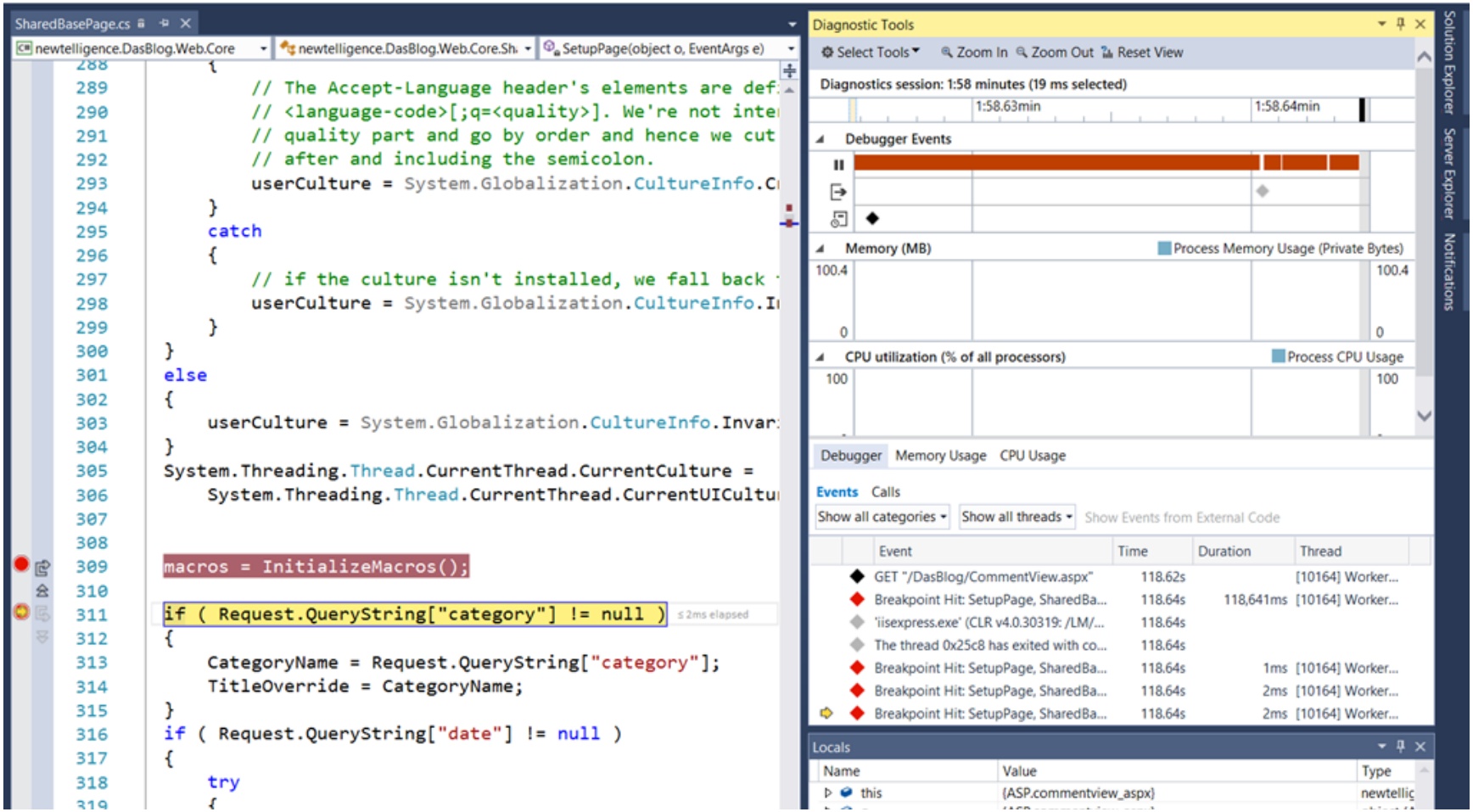
ROS



なぜROSを利用するか

- ミドルウェアだけでなく、オープンプラットホーム
  - 他者の手を無料で借りる
  - 独自で開発すると手が届かない機能は様々
- 開発をサポートする機能
  - Introspection
  - デバッグ
  - ソース再利用
  - …
- ツールは豊か
  - ロボット開発にいいツールは必須
  - ツールの例：デバッグツール

デバッグ：PC


SharedBasePage.cs
newtelligence.DasBlog.Web.Core
newtelligence.DasBlog.Web.Core.Sh
SetupPage(object o, EventArgs e)
// The Accept-Language header's elements are def
// <language-code>[;q=<quality>]. We're not inte
// quality part and go by order and hence we cut
// after and including the semicolon.
userCulture = System.Globalization.CultureInfo.C
catch
// if the culture isn't installed, we fall back
userCulture = System.Globalization.CultureInfo.I
else
userCulture = System.Globalization.CultureInfo.Invar
System.Threading.Thread.CurrentThread.CurrentCulture =
System.Threading.Thread.CurrentThread.CurrentUICultu
macros = InitializeMacros();
if (Request.QueryString["category"] != null)
≤ 2ms elapsed
CategoryName = Request.QueryString["category"];
TitleOverride = CategoryName;
if (Request.QueryString["date"] != null)
try
Diagnostic Tools
Select Tools
Zoom In
Zoom Out
Reset View
Diagnostics session: 1:58 minutes (19 ms selected)
1:58.63min
1:58.64min
Debugger Events
Memory (MB)
Process Memory Usage (Private Bytes)
100.4
0
CPU utilization (% of all processors)
Process CPU Usage
100
Debugger
Memory Usage
CPU Usage
Events
Calls
Show all categories
Show all threads
Show Events from External Code
Event
Time
Duration
Thread
GET "/DasBlog/CommentView.aspx"
118.62s
[10164] Worker...
Breakpoint Hit: SetupPage, SharedBa...
118.64s
118,641ms
'iisexpress.exe' (CLR v4.0.30319: /LM/...
The thread 0x25c8 has exited with co...
1ms
2ms
Locals
Name
Value
Type
this
{ASP.commentview_aspx}
Solution Explorer
Server Explorer
Notifications

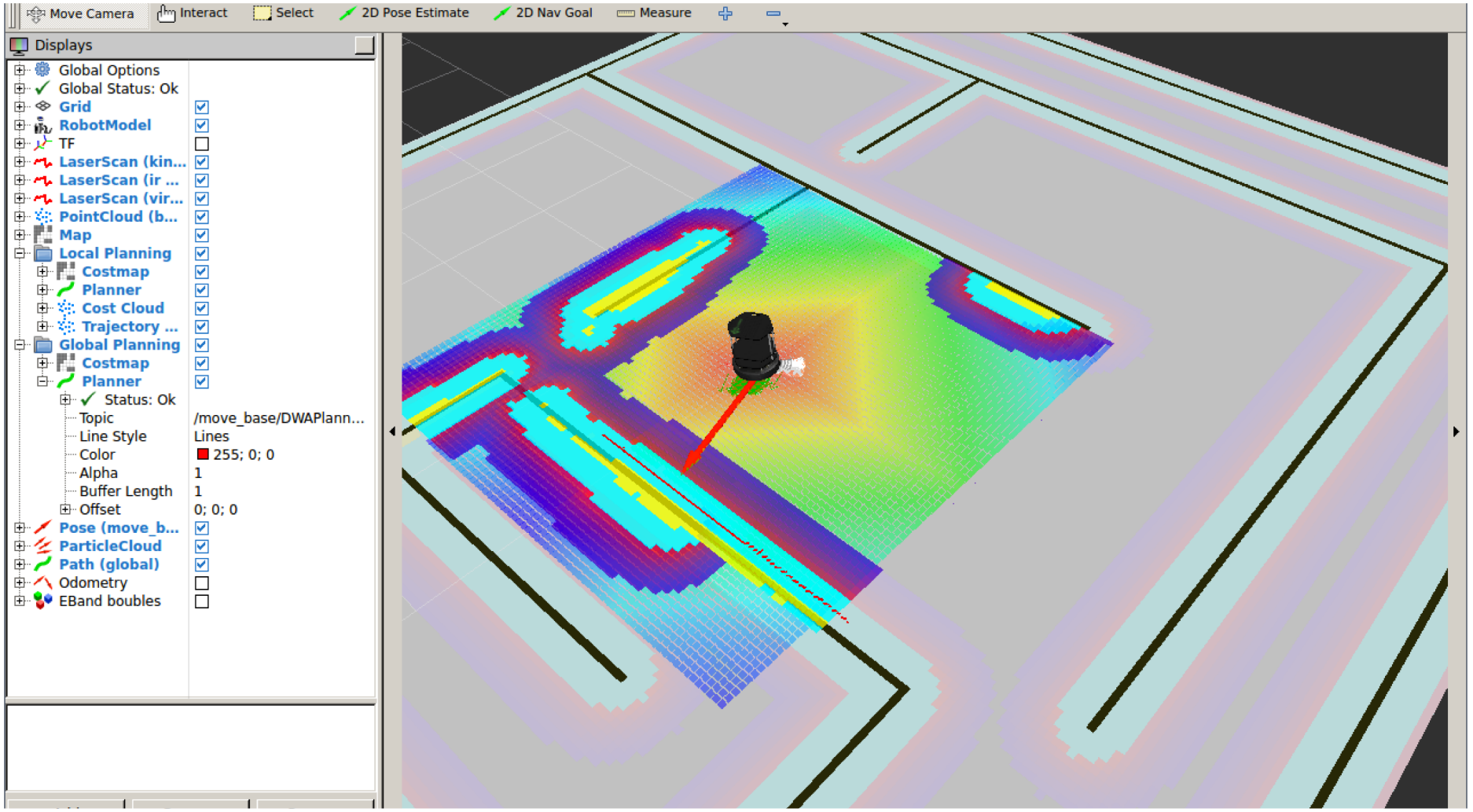
デバッグ：ロボット

0.47047596160651484, 0.012463983269873857, 0.13185027912023128, 0.5380365386000314, 0.403864612045705,
0.06069930337048779, 0.3691795213234019, 0.8515692765843788, 0.0736967442312384, 0.2843486311730886,
0.8538529785225724, 0.3129984654314725, 0.3584207454827225, 0.4537846637411378, 0.7561766125048414,
0.8508870244042188, 0.0520696074562399, 0.8765451038148542, 0.15299405008975697, 0.5867355890849806,
0.9669883187750915, 0.41628598565655384, 0.34816285093525845, 0.9728408650406579, 0.41829494089973274,
0.579997374892018, 0.5531830445281459, 0.24530608285472844, 0.38526280587020356, 0.2210817119943037,
0.07975484090387797, 0.6433789010590776, 0.14258622222962336, 0.47068371292023214, 0.021146762861280366,
0.7809218654795851, 0.059743545404254084, 0.7174453572838055, 0.11796690651681152, 0.45170824264501075,
0.3463075513634315, 0.12329222545582419, 0.9238715097652316, 0.20246585186042165, 0.07287853589316085,
0.8857115222065027, 0.2607754782454512, 0.7871998842710477, 0.3123418728541889, 0.24793655686384475,
0.03936840867259228, 0.15830275932784887, 0.0588537758302089, 0.6905061474343309, 0.45187144905773735,
0.3137586806320821, 0.3928776902569401, 0.18194845109045765, 0.7772116865207355, 0.10878199902185848,
0.6046192473442614, 0.3783574047058046, 0.9335545367770111, 0.5931631033298297, 0.8227864249862641,
0.37984882478112314, 0.35863612685228197, 0.712004279109606, 0.3414544916392299, 0.4118358463966818,
0.709283617996954, 0.14020958524673632, 0.03867697529807812, 0.5240273928062406, 0.631862692442692,
0.9698634206694162, 0.6034612197461879, 0.3218208000413312, 0.6832981693054933, 0.6050553280734954,
0.6604449894889293, 0.3442873321468821, 0.18196942376936986, 0.4199897381622488, 0.8804476148633672,
0.1922718532155836, 0.963177730081603, 0.21728977473403277, 0.20754416331944092, 0.30192492471822563,
0.19063402733809054, 0.9498797020928332, 0.6673141005665001, 0.45745969941649967, 0.4351040209987119,
0.5928193275952122, 0.6823498640502099, 0.7442921879931558, 0.009326500338331 49, 0.11983479696483657,
0.0519672566623508, 0.36971891569137594, 0.190717653457389, 0.7492902839065593, 0.5602249232567452,
0.6585780171293569, 0.08658888622314564, 0.6945120787132953, 0.5580665700278967, 0.48146812759625746,
0.4540375145339196, 0.8059387960368088, 0.8555468352647819, 0.5737110434503095, 0.35666621246462693,
0.06659849272034857, 0.21051537392984276, 0.731280246174109, 0.9151365268872526, 0.37390717986528854,
0.32541148512546336, 0.5594068298257496, 0.08044184535792642, 0.5591207304751257, 0.20915450867051832,
0.02534903593989679, 0.19416932780283314, 0.34906251623185813, 0.8432152857763088, 0.8664024388302471,
0.2840496778118137, 0.2781658794647851, 0.2465128654531562, 0.9988375836561965, 0.887499520253597,
0.027578461404808352, 0.29371750902319593, 0.8873891214375212, 0.914573150995103, 0.000652243172693745,
0.06772123956909537, 0.9597290829459296, 0.7714181981544204, 0.32507290632274355, 0.11058804894490903,
0.9998152138773576, 0.622944943150389, 0.3816913626678231, 0.7083424647825276, 0.11404328136373076,
0.21818485912970653, 0.08168144717155512, 0.6732894753601069, 0.5991484405675498, 0.026001431290659682,
0.7854449857322512, 0.2454220869235253, 0.9264245146883056, 0.9350749518287534, 0.9652777708676377,
0.8655942624669641, 0.27395894599419734, 0.0785308404808881, 0.3250820262765106, 0.5039220523760805,
0.4985074812618765, 0.01020455824663058, 0.5881052368037296, 0.017811209585336 22, 0.27810744561068323,
0.4818559083236109, 0.535352899944137, 0.6768776195049253, 0.165449685238322 15, 0.5330947114782651,
0.6964628631194874, 0.09006024022384884, 0.07967584234804681, 0.9611481706094649, 0.9264391787583394,
0.6569180180956492, 0.9620719195023297, 0.4245900557129376, 0.7340280656377078

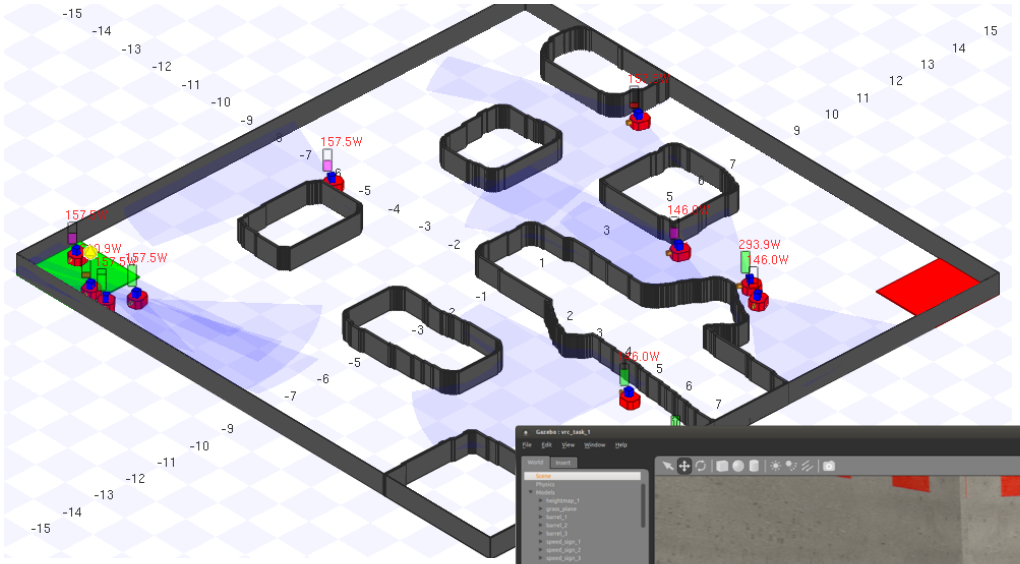
デバッグ

- ロボット開発で特に難しいところ
  - データが大量
  - 世界はポーズできない
- 世界をロボットのように見えると問題の原因が明確になる
- ライブまたは再生で見る
- 1フレームづつで見ると終わらない

デバッグ：データの可視化



デバッグ：シミュレータ



ロボットの周りの環境も管理できる
→テストしやすい
→繰り返しやすい

Part 3

ROS 2への道

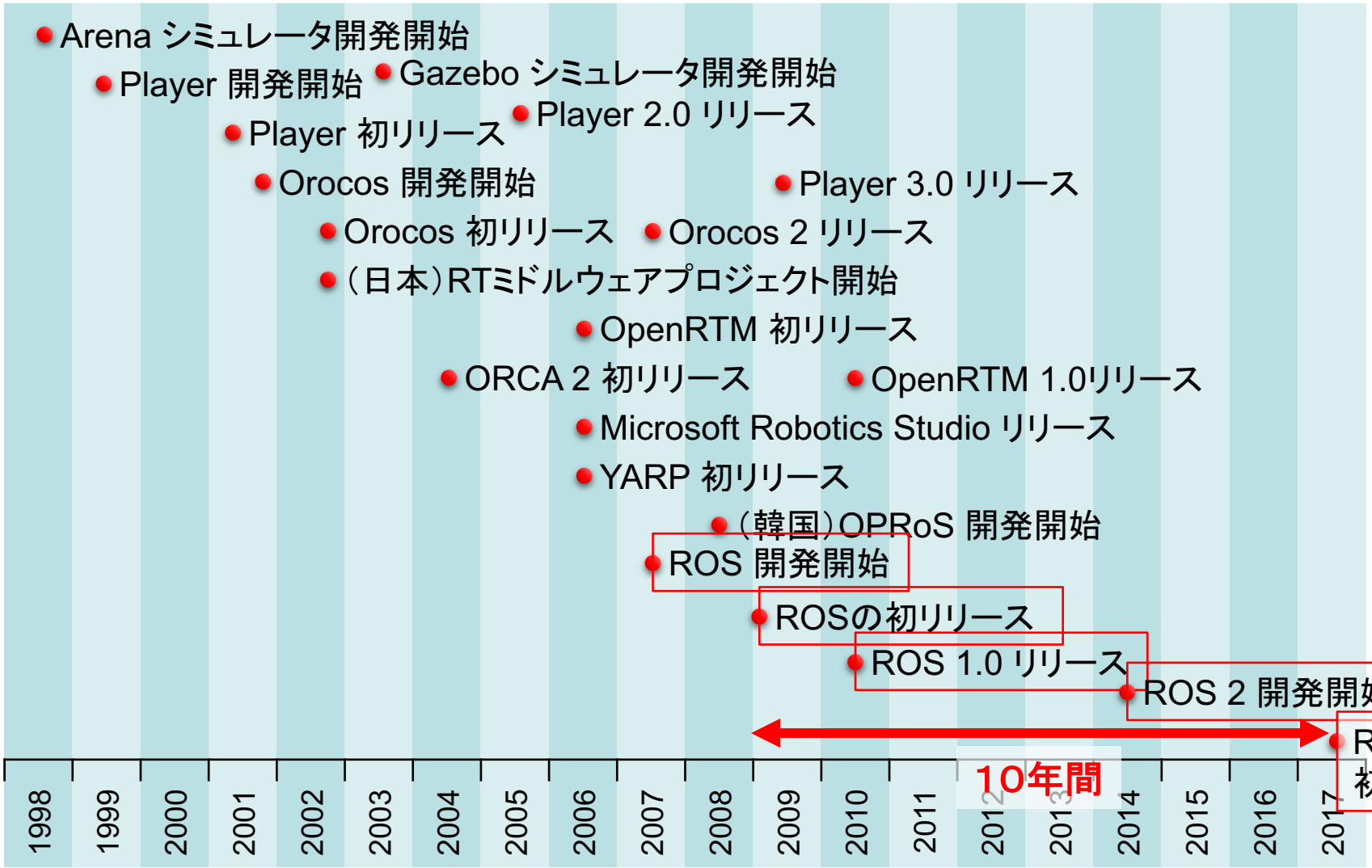
ROSは・・・

- ロボット用のソフトウェアプラットフォーム
- ロボット分野で大成功
  - 様々なロボットで利用されている
  - 研究分野で人気である
  - ベンチャー会社でも大手メーカーでも利用されている
  - 購入可能な製品もある

もうROS 2?

- ROS 1がある!
- ROS 1は広く利用されている!
- ROS 1ようのソフトウェアはまだ増えている!
- ROS 1を使い始めたばかりだ!
- ROS 1の利用するコミュニティはやっと大きくなり始めた！

ロボットソフトウェアの歴史



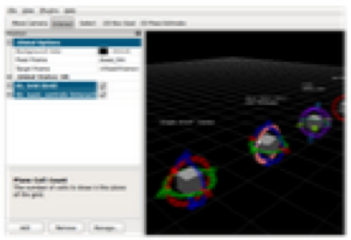
ROS

- Robot Operating System
- 2007年から
- フレームワークだけではなくて、<u>プラットフォーム全体</u>

Plumbing

コンポーネント
間の通信

+



Tools

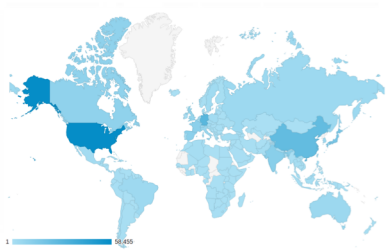
ツール

+

Capabilities

頻繁に利用される
ロボットの機能

+



Ecosystem

エコシステムと
コミュニティー

ROS

ROS 2活動の中心

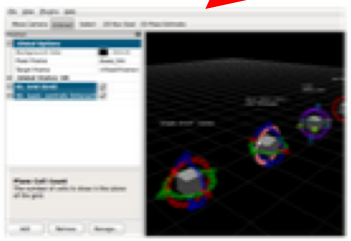
- 2007年から
- フレームワーク
  プラットフォーム全体

主にポーティングと改善

Plumbing

コンポーネント
間の通信

+



Tools

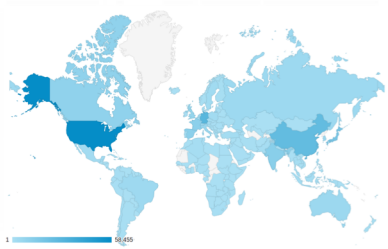
ツール

+

Capabilities

頻繁に利用される
ロボットの機能

+



Ecosystem

エコシステムと
コミュニティー

ロボットソフトウェアの特質

- 複数の分野の混合
  - 制御理論、信号処理、自動計画、AI、…
- リアルタイムと非リアルタイムの混合
- 大量のデータ処理
- CPU、メモリは甚だしく利用も、
  小さなマイコン利用
- ハードウェアとソフトウェア
  アーキテクチャーの複雑さ
- 現実世界との複雑なインタラクション
- …

ロボットソフトウェアの特質

- 複数の分野の混合
  - 制御理論、信号処理、自動計画、AI、…
- リアルタイムと非リアルタイムの混合
- 大量のデータ処理
- CPU、メモリは甚だしく利用も、小さなマイコン利用
- ハードウェアとソフトウェアアーキテクチャーの複雑さ
- 現実世界との複雑なインタラクション
- …

企業にとってROSの問題

- OSRFからのソフトウェア・ノード以外信頼できない

○

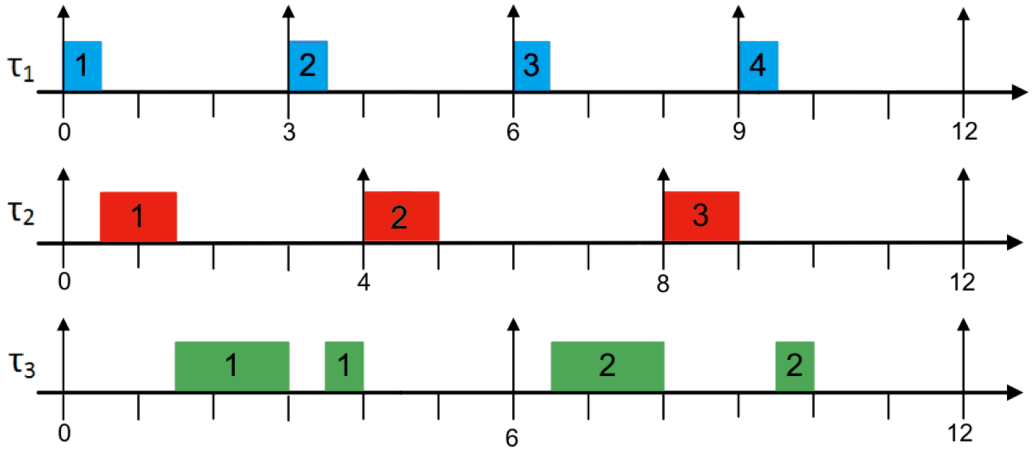
×

企業にとってROSの問題

- 完全自家製技術だから信頼性が低い
  - 特に通信プロトコル
- 認証不可能（コアライブラリでも）
  - 安全システム・高信頼システムで利用不可

企業にとってROSの問題

- リアルタイムが難しい
- モータードライバーや決定的なシステムにROSが利用できない



企業にとってROSの問題

- POSIX型OSだけ
- Ubuntuではないと使いにくい
- 組み込みシステムは不可能

ubuntu

redhat LINUX

CentOS

debian

○

×

解決策： ROS 2

- コアをゼロから開発しなおすこと
- ROS1ができないことがコアユーズケース
  - 自家製ソフトウェアの減少
  - 組み込み（特に小リソースマイコン）
  - リアルタイム
  - 決定的実行（ノードライフサイクル管理等）
  - インフラの一貫性、規格化
  - 管理された・記録された開発プロセス

Part 4

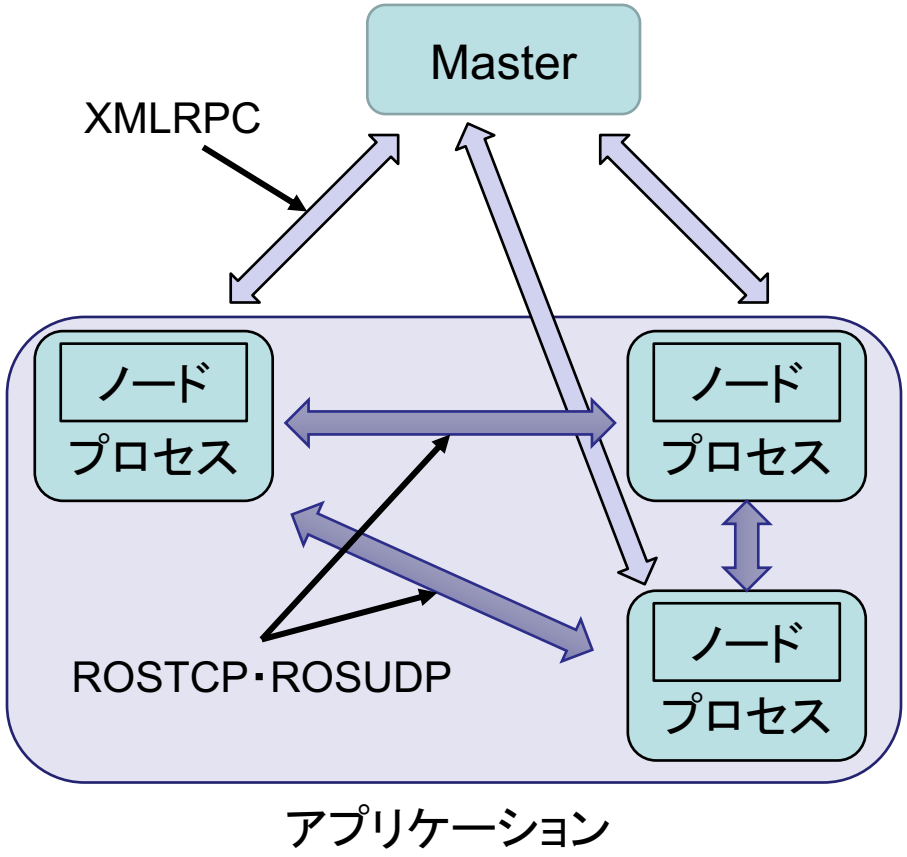
ROS 1とROS 2の差

ROS 1とROS 2の差

- 新しい通信プロトコル
- 通信プロトコルの実装は入れ替え可能
- QoSが管理可能
- ライブラリアーキテクチャの更新
- 全スタックがリアルタイム可能
- ノード構造の更新
- OSサポートの拡大
- 開発プロセスの記録
- 安全認証取得可能

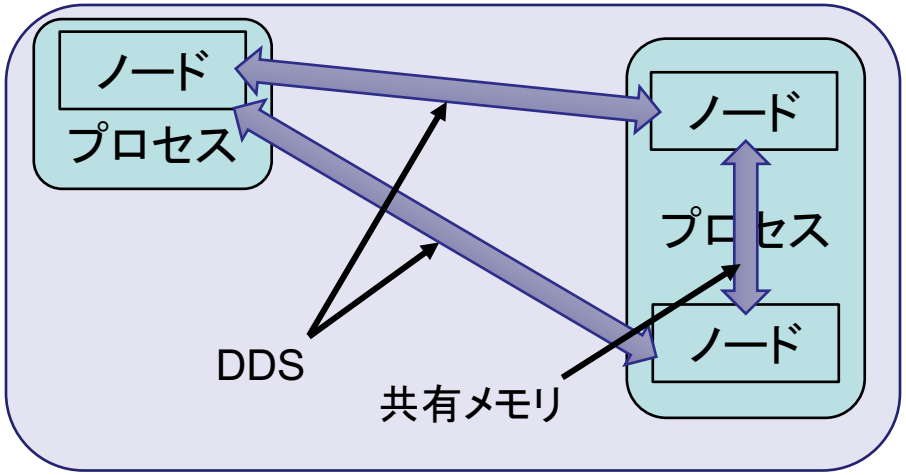
ROS 1アプリケーションの構造

- ノードが別々のプロセスで実行中
- Master（roscore）が必要
  - パラメータサーバ等も
- データ通信はROSTCPで
- メタデータ通信はXMLRPCで

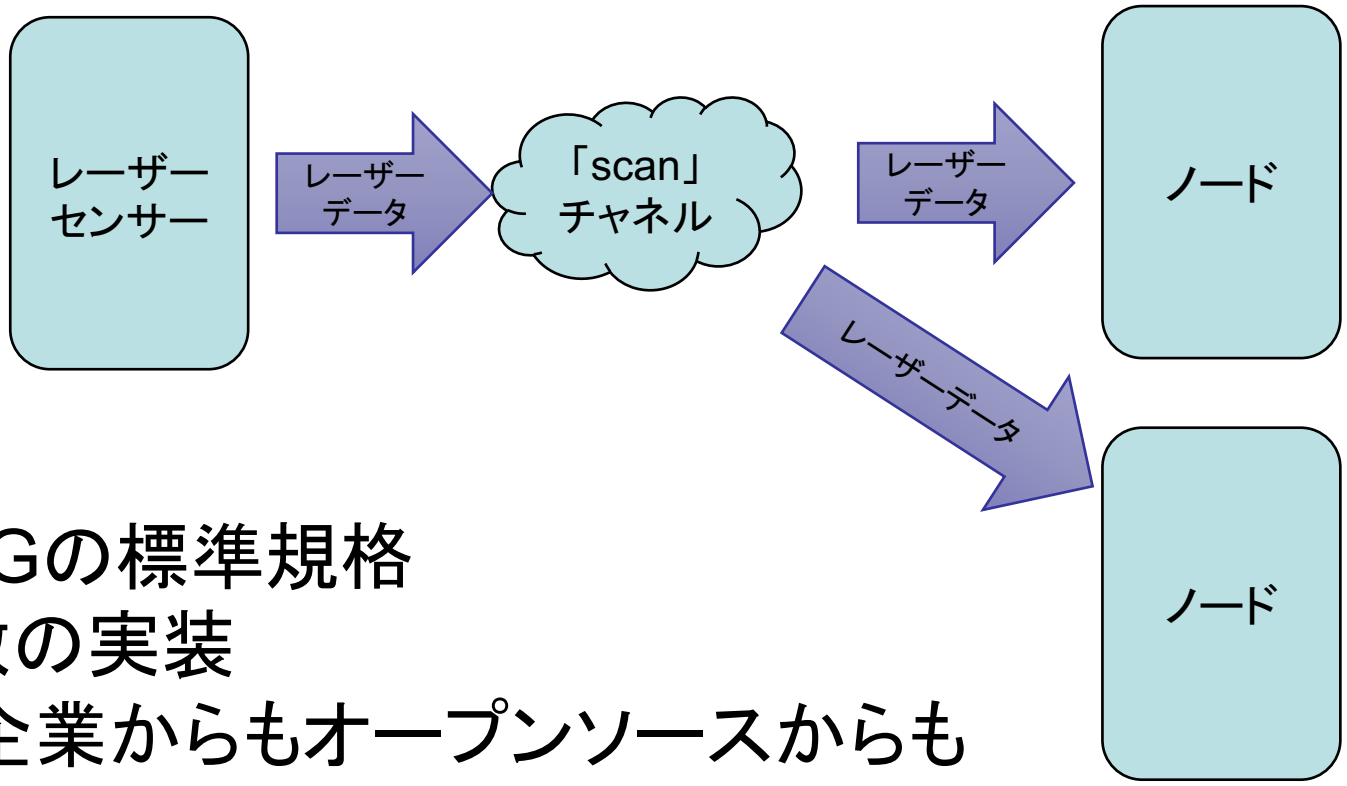


ROS 2アプリケーションの構造

- ノードがシステムのニーズに合わせて別々のプロセスでも同一プロセスでも実行可能
- Masterがない
  - Single point of failureがなくなった
- データ通信もメタデータ通信もDDSや共有メモリで
  - 自家製ではない

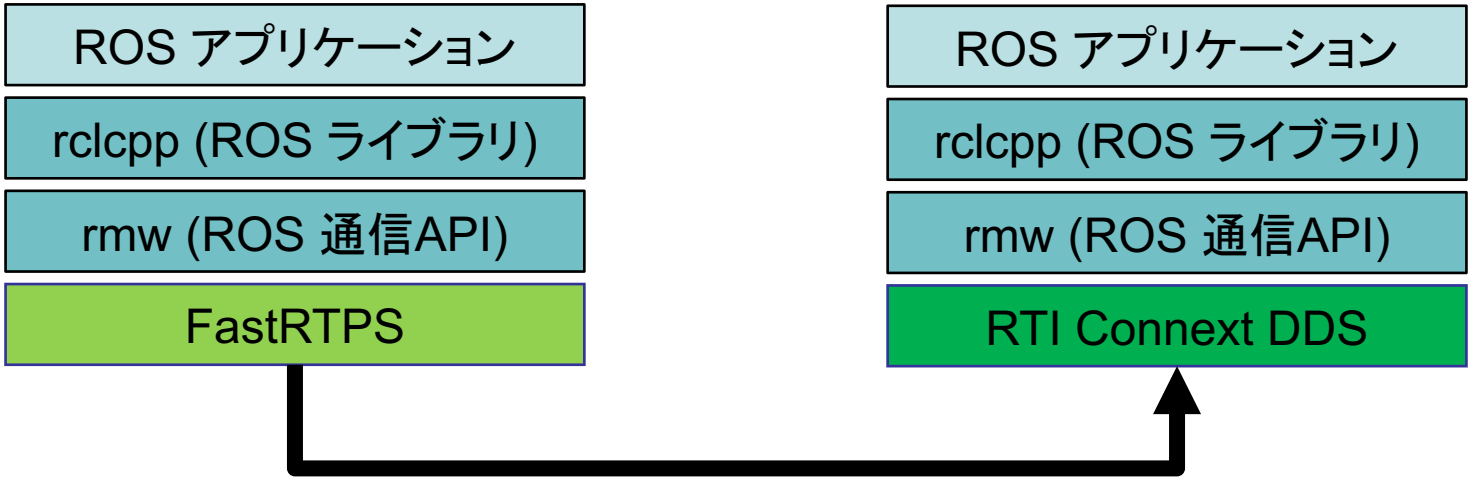


新しい通信ミドルウェア：DDS (Data Distribution Service)



- OMGの標準規格
- 複数の実装
  - 企業からもオープンソースからも
- 安全認証版

DDS実装間の互換性



サポートありDDS ベンダー	ライセンス
eProsima FastRTPS	Apache 2
RTI Connext DDS	Commercial
ADLink OpenSplice DDS	Commercial

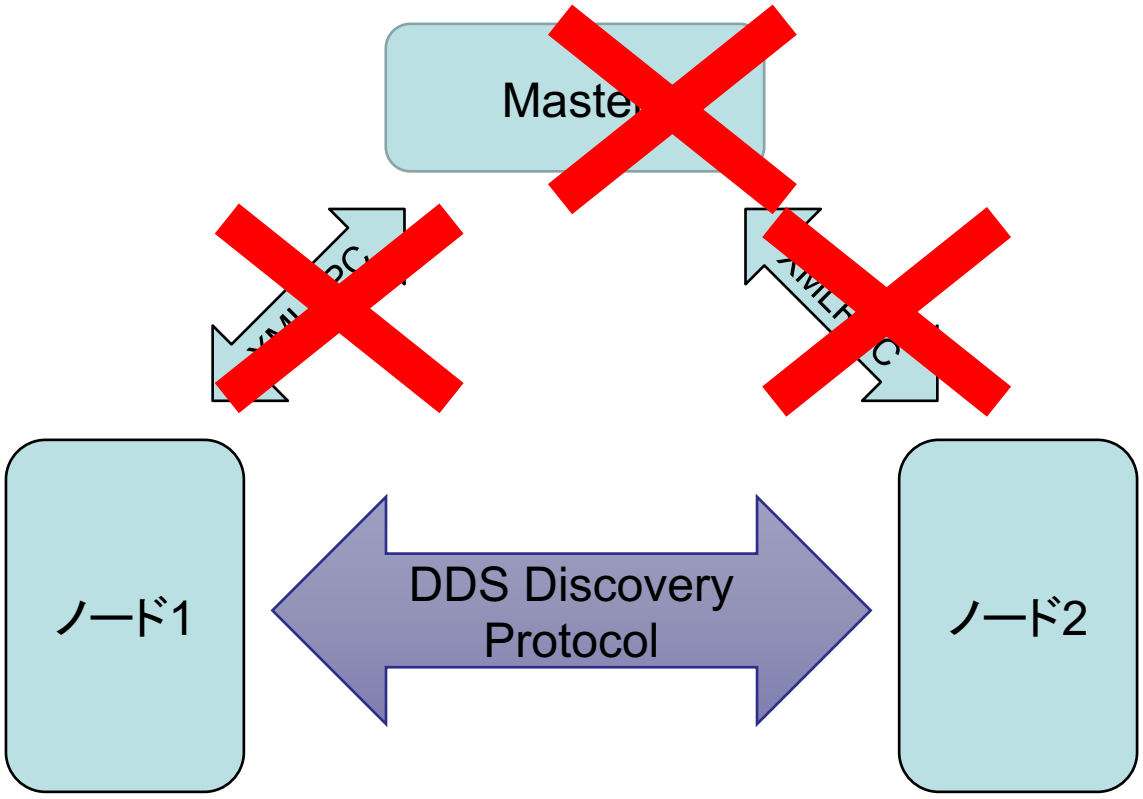
Open Source Robotics Foundation

DDSのQoS設定

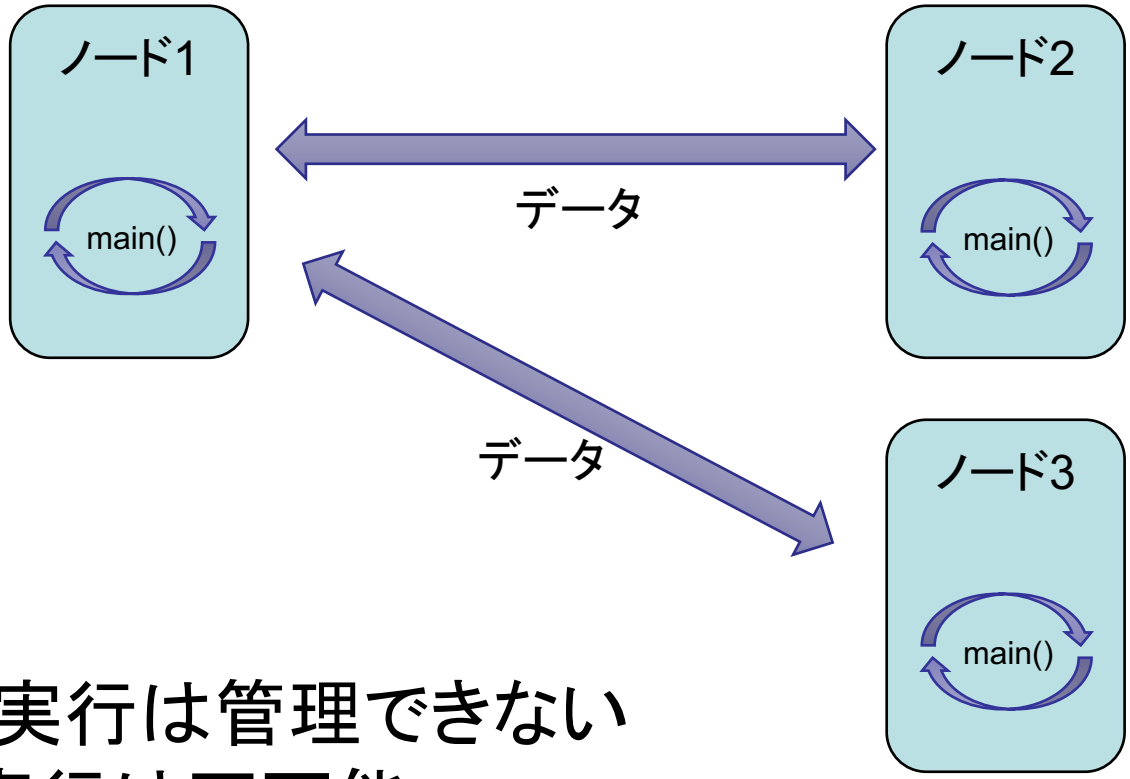
ROS1:
チューニングができない

ROS 2/DDS:
細かいチューニング可能

Masterの排除

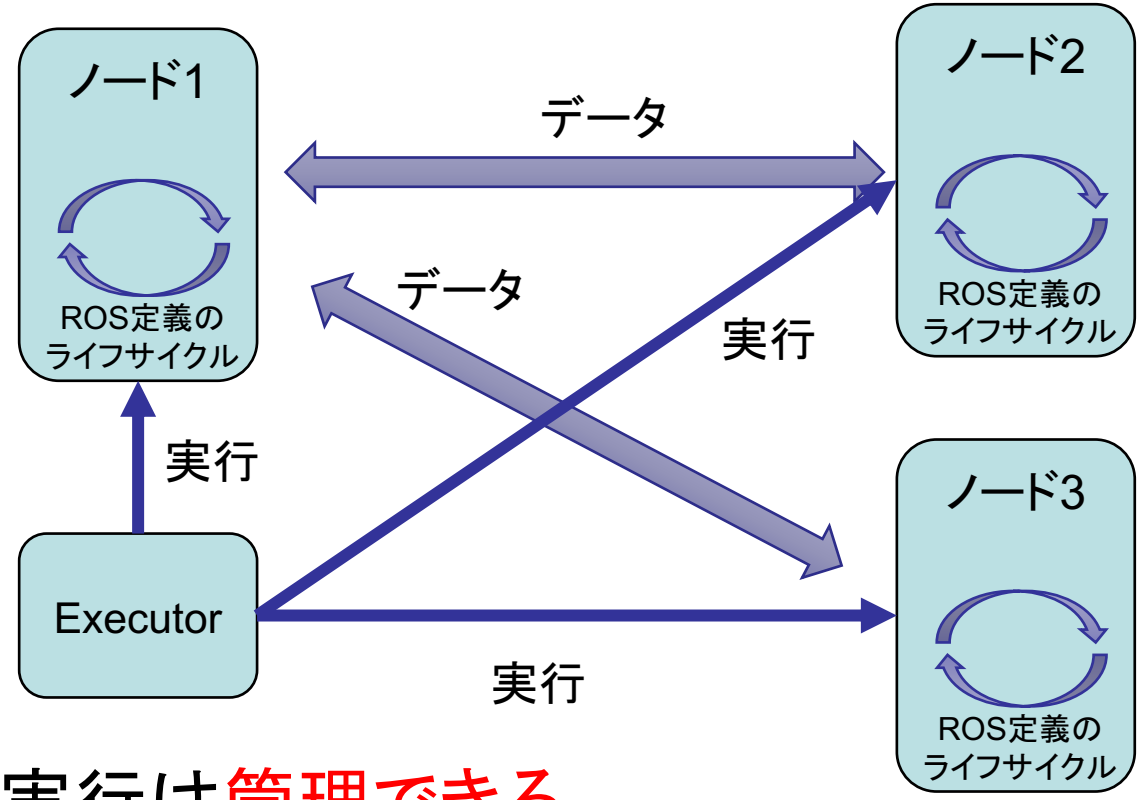


ノードライフサイクル – ROS 1



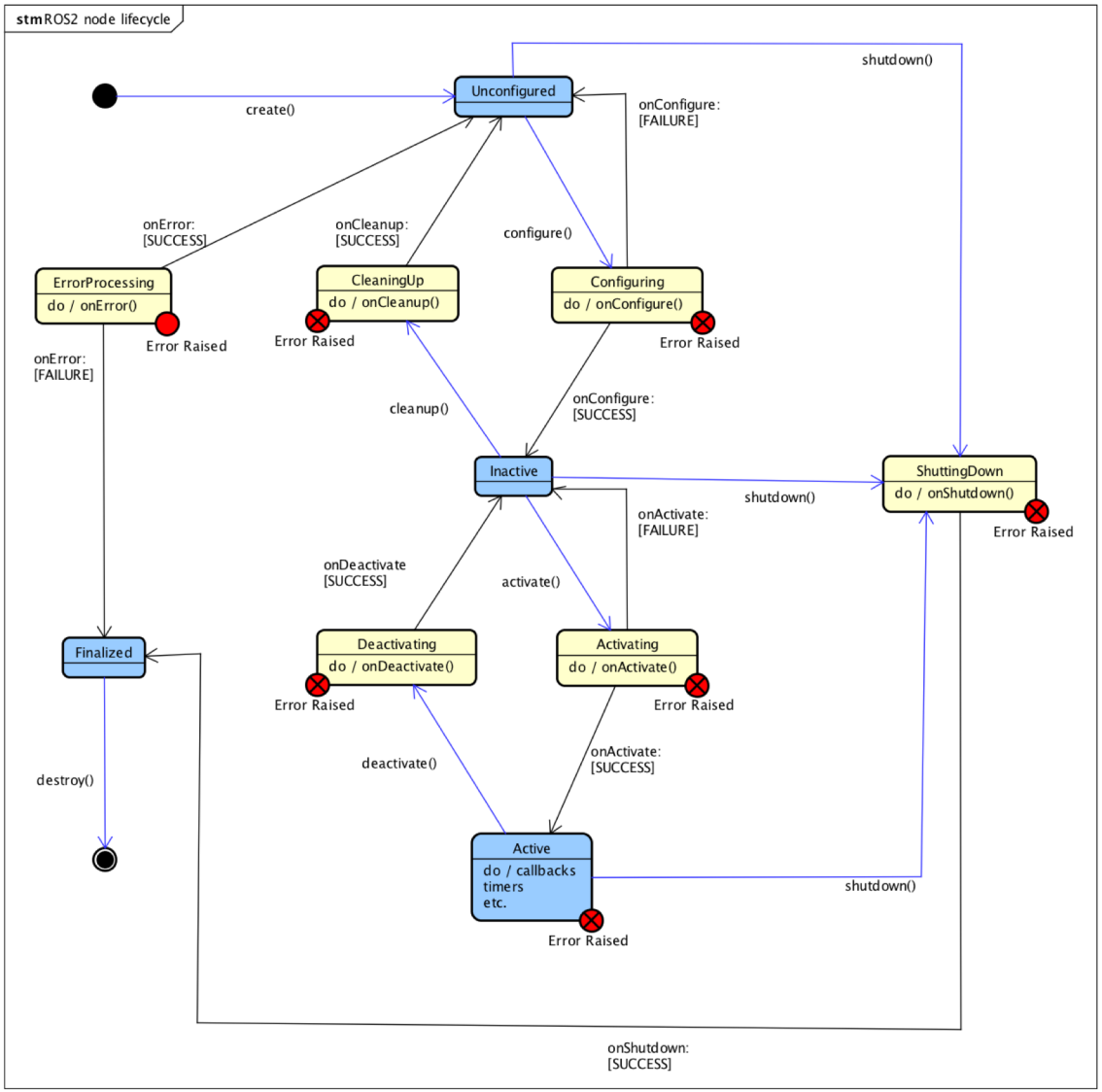
ノードの実行は管理できない
決定的実行は不可能

ノードライフサイクル – ROS 2

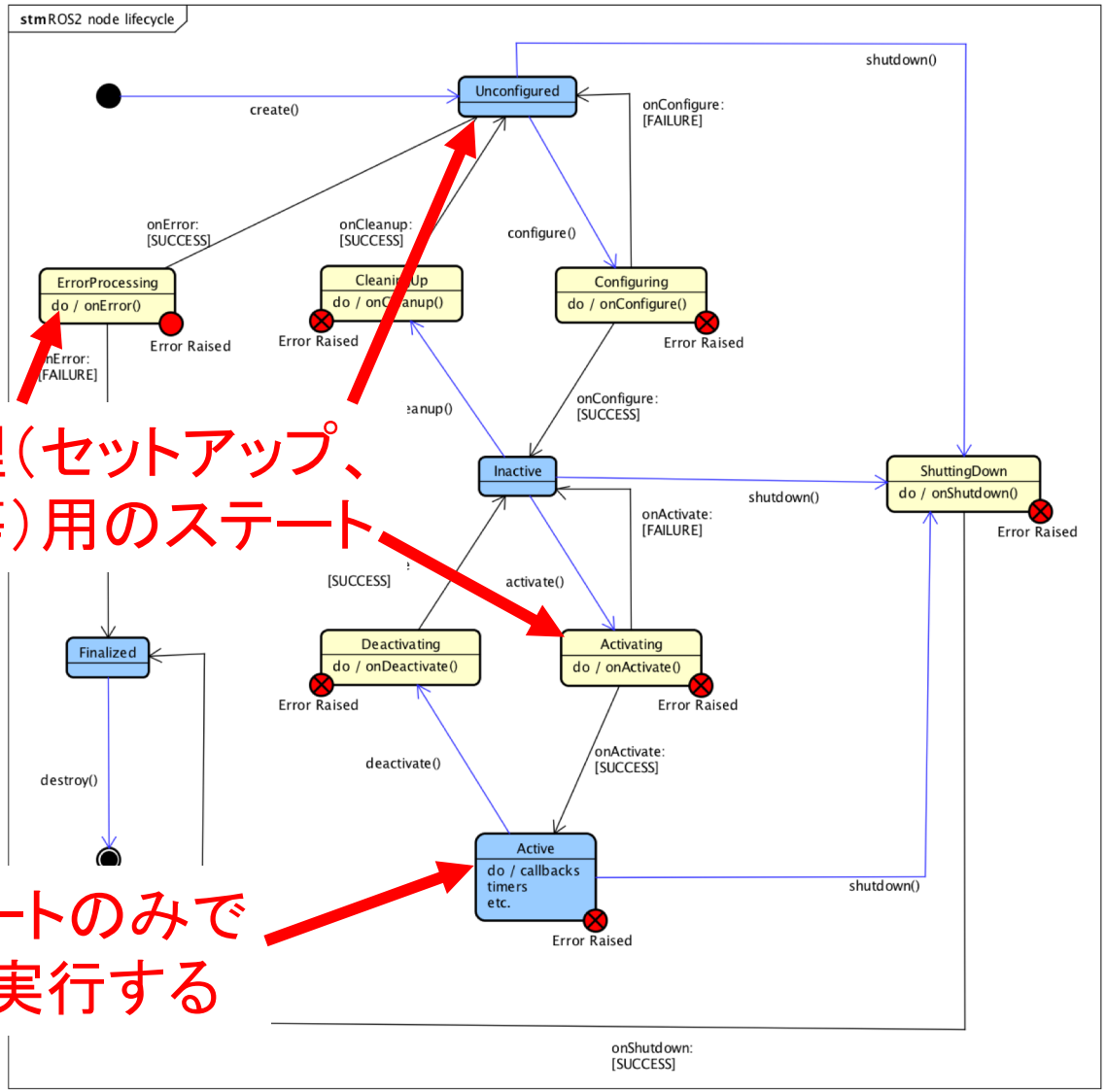


ノードの実行は管理できる
決定的実行は可能

ノードライフサイクル – ROS 2



ノードライフサイクル – ROS 2



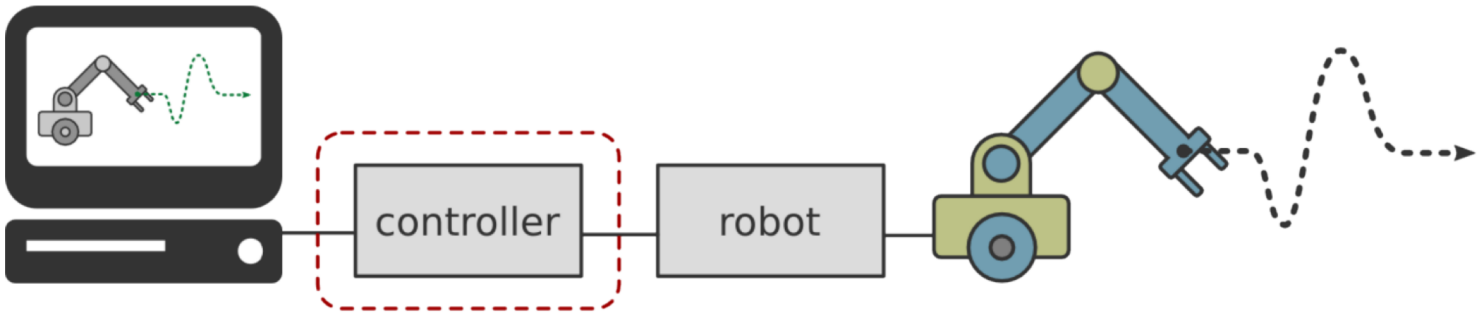
システム管理（セットアップ、エラー状態等）用のステート

このステートのみで本機能を実行する

ノードライフサイクル – ROS 2

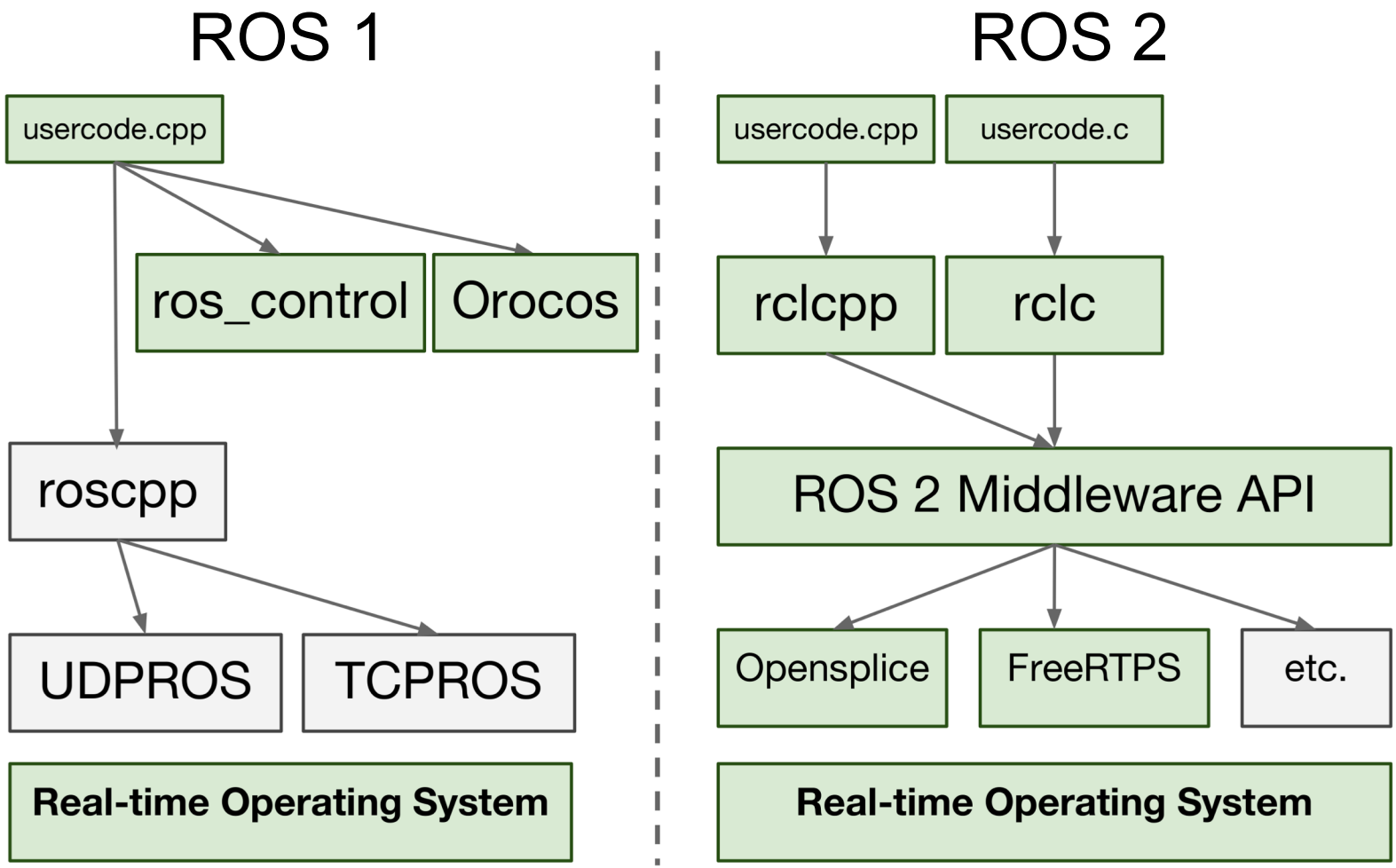
- もしろん、メーン関数でノードの実装はまだ可能!
  - –（プロトタイピングに必須だから）

リアルタイム可能



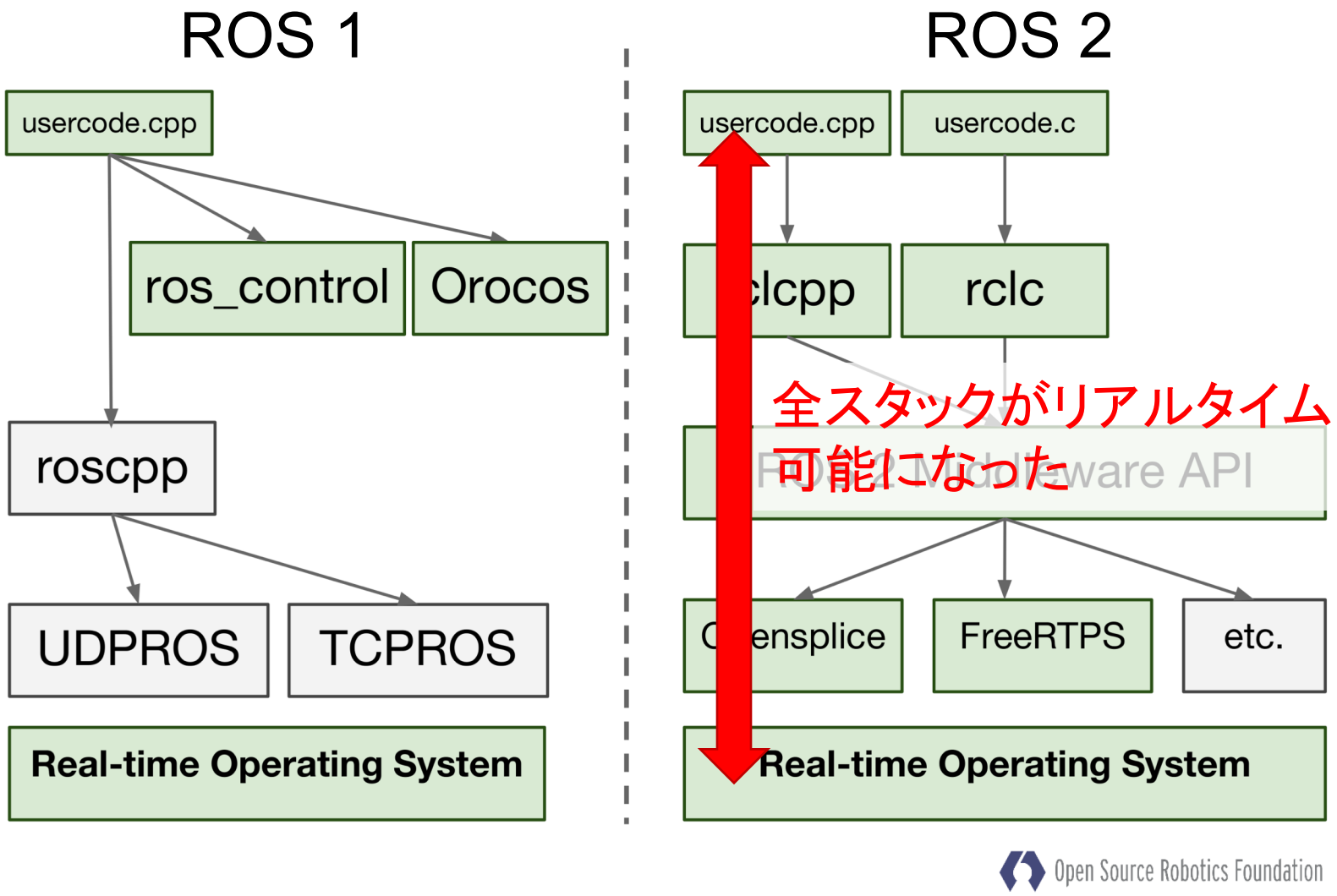
Open Source Robotics Foundation

リアルタイムアーキテクチャー

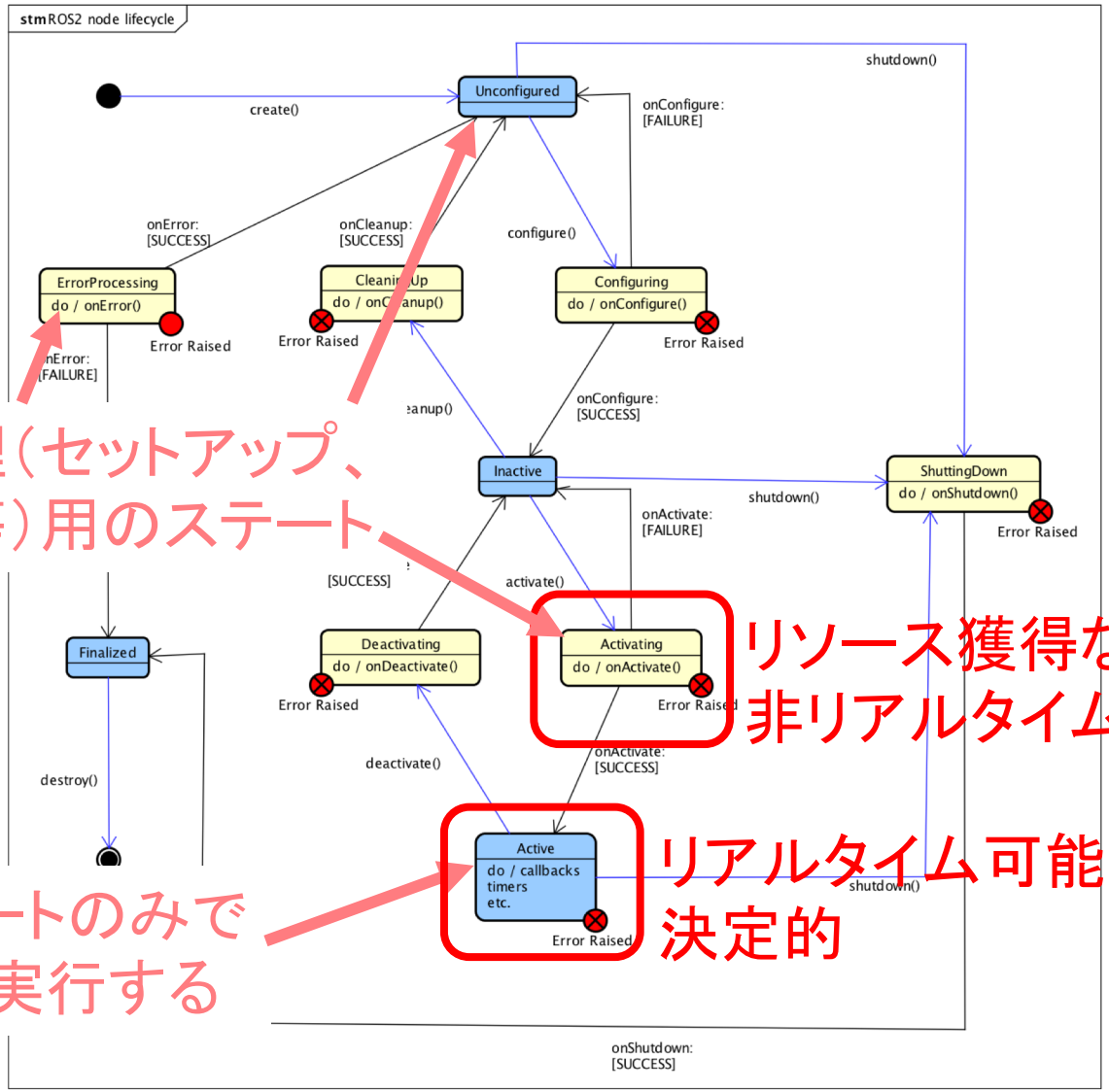


Open Source Robotics Foundation

リアルタイムアーキテクチャー



ノードライフサイクル – ROS 2



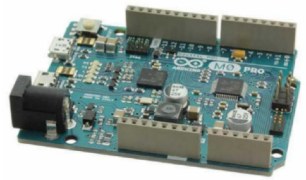
システム管理（セットアップ、エラー状態等）用のステート

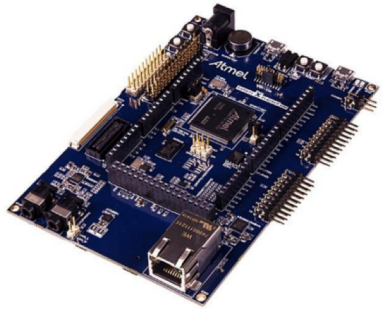
リソース獲得などの非リアルタイムコード

リアルタイム可能決定的

このステートのみで本機能を実行する

組み込みROS





	"small" 32-bit MCU	"big" 32-bit MCU
Core	ARM Cortex-M0	ARM Cortex-M7
Speed	48 Mhz	300 Mhz
RAM	32 KB	384 KB
Flash	256 KB	2048 KB
Cost @ 1K units	$2	$10
Comms	USB FS	Ethernet, USB HS

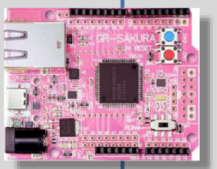
Open Source Robotics Foundation

組み込みROS

ROS2 Implementation to MCU DEMO

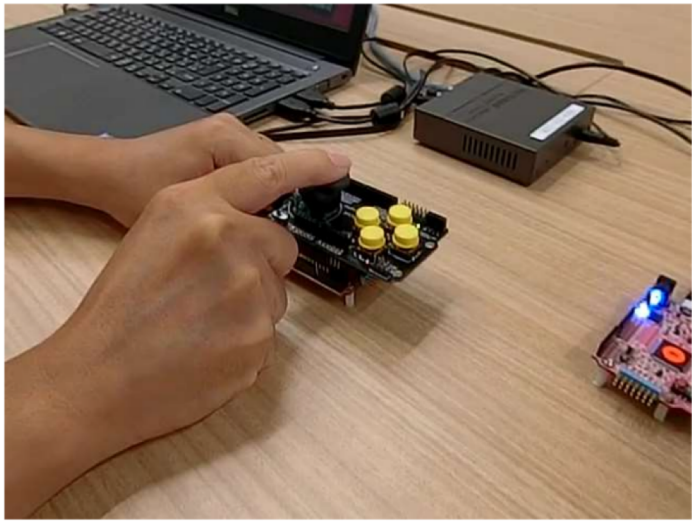
Sensor node
(publisher)

Joystick shield



GR-SAKURA II

FreeRTPS



Actuator node
(subscriber)

Pan-tilt
unit

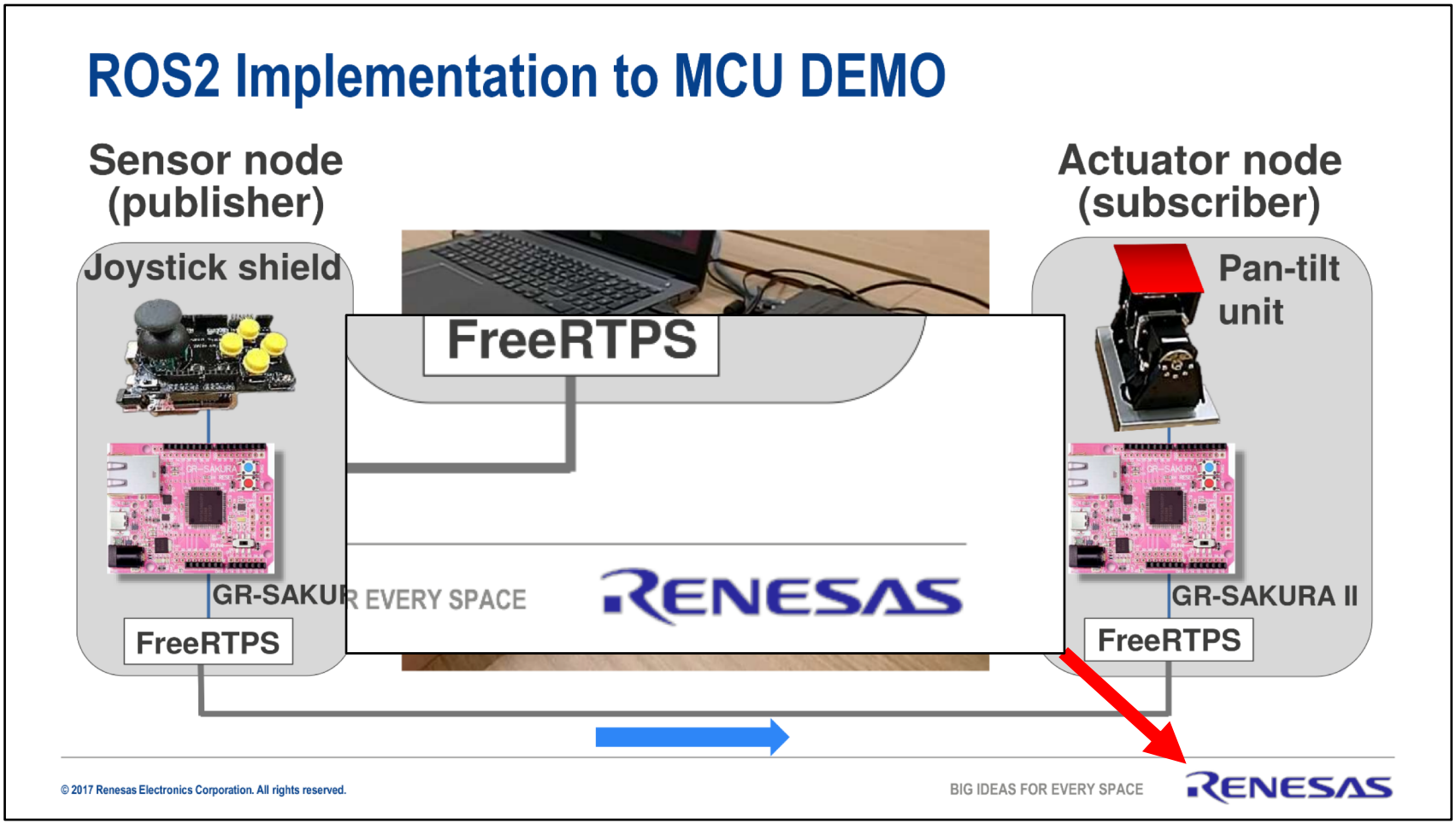
GR-SAKURA II

FreeRTPS

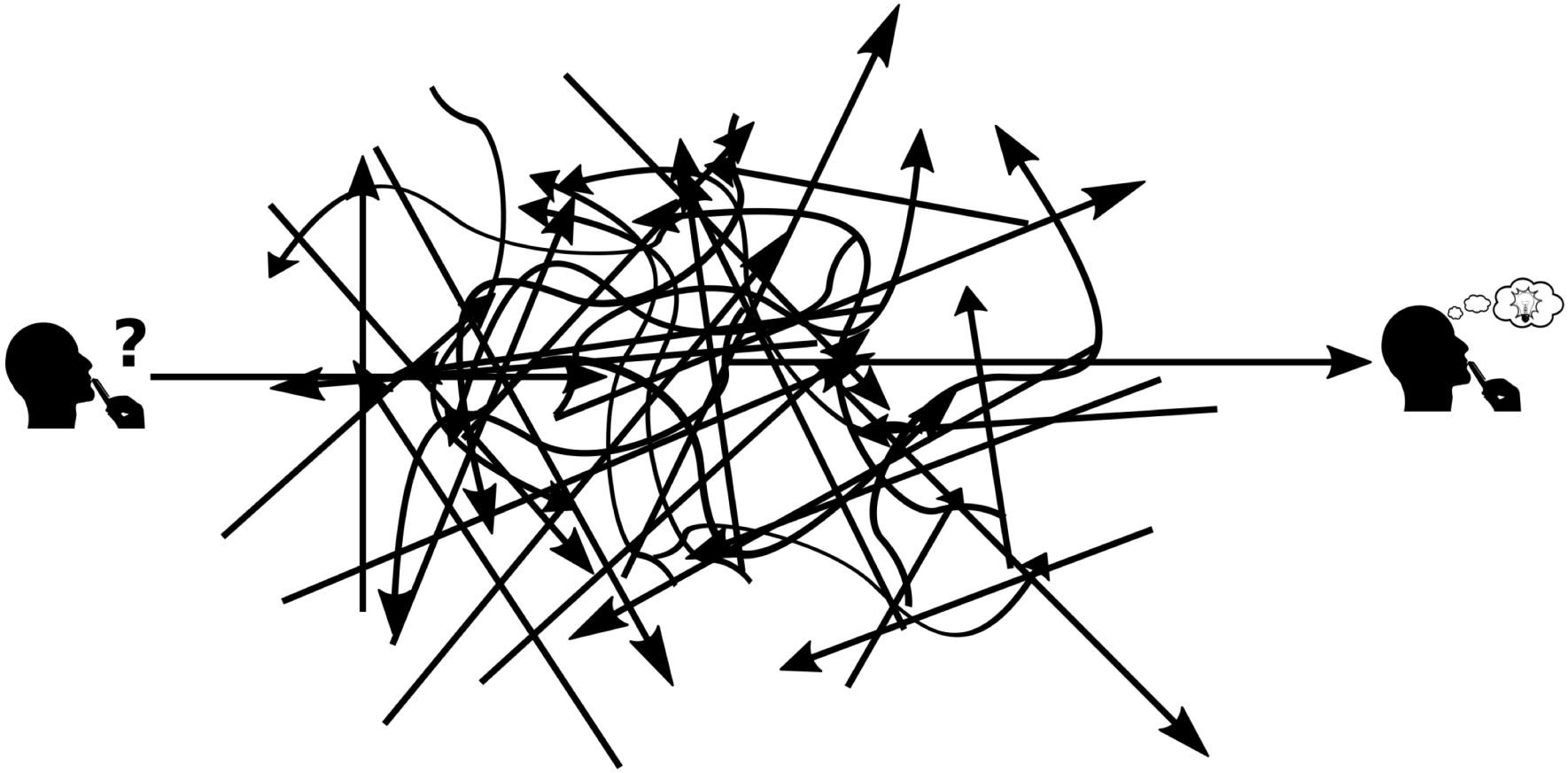
BIG IDEAS FOR EVERY SPACE

RENESAS

組み込みROS

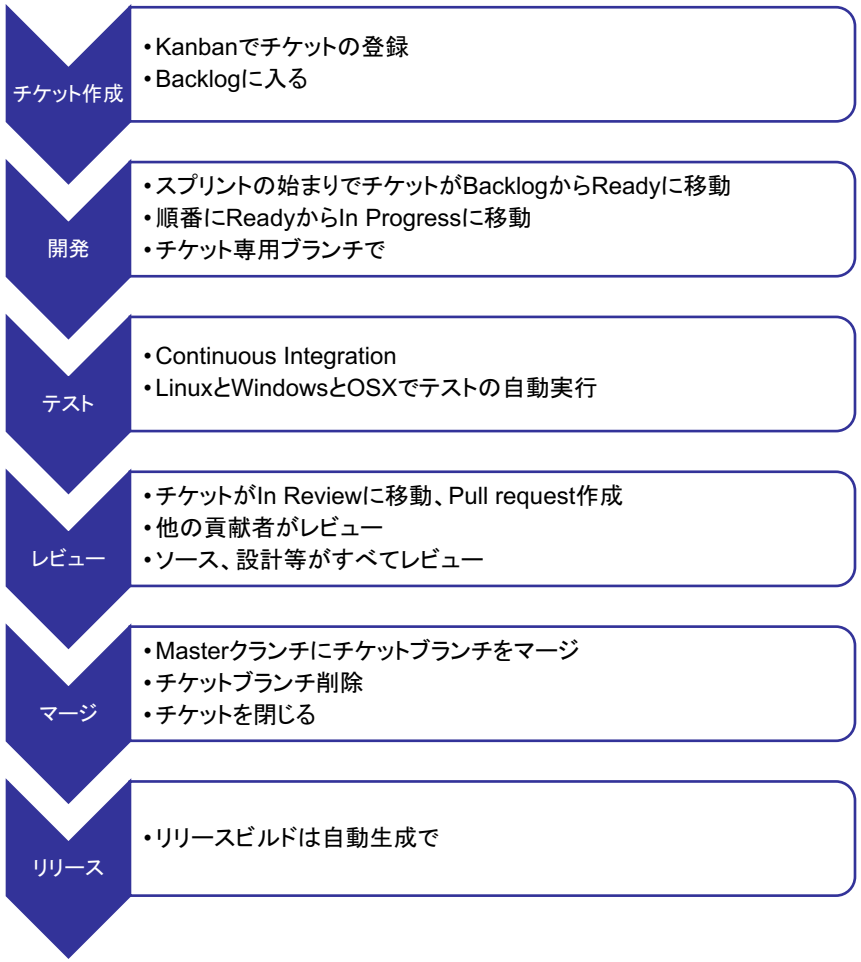

ROS2 Implementation to MCU DEMO
Sensor node
(publisher)
Actuator node
(subscriber)
Joystick shield
Pan-tilt
unit
FreeRTPS
GR-SAKURA
GR-SAKURA II
FreeRTPS
FreeRTPS
EVERY SPACE
RENESAS
© 2017 Renesas Electronics Corporation. All rights reserved.
BIG IDEAS FOR EVERY SPACE
RENESAS

ROS1の開発プロセス

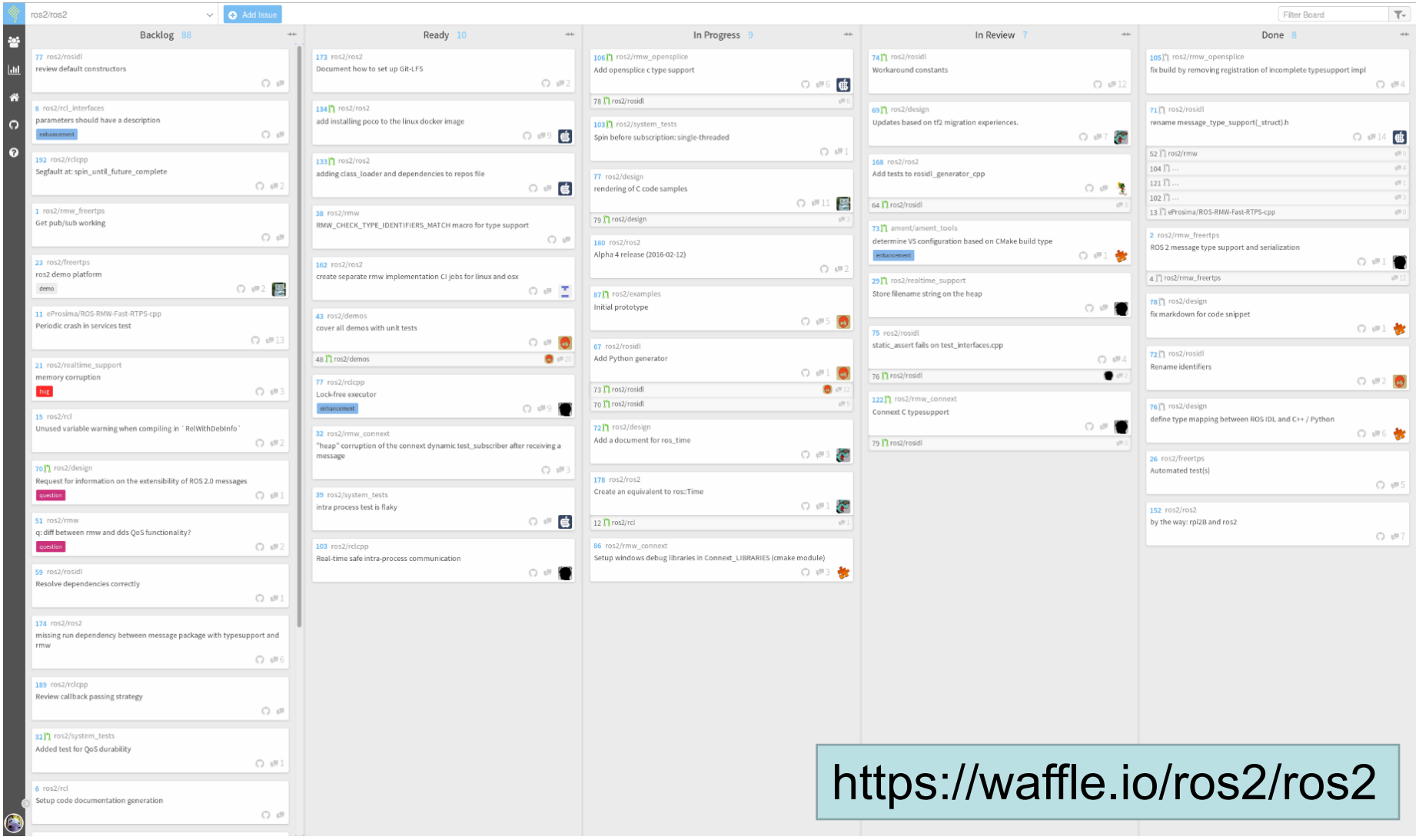


ROS2の開発プロセス

- アジャイル
- Test-Driven（TDD）
- カンバンで管理
- Linux、Mac OSX、WindowsでCIとテスト

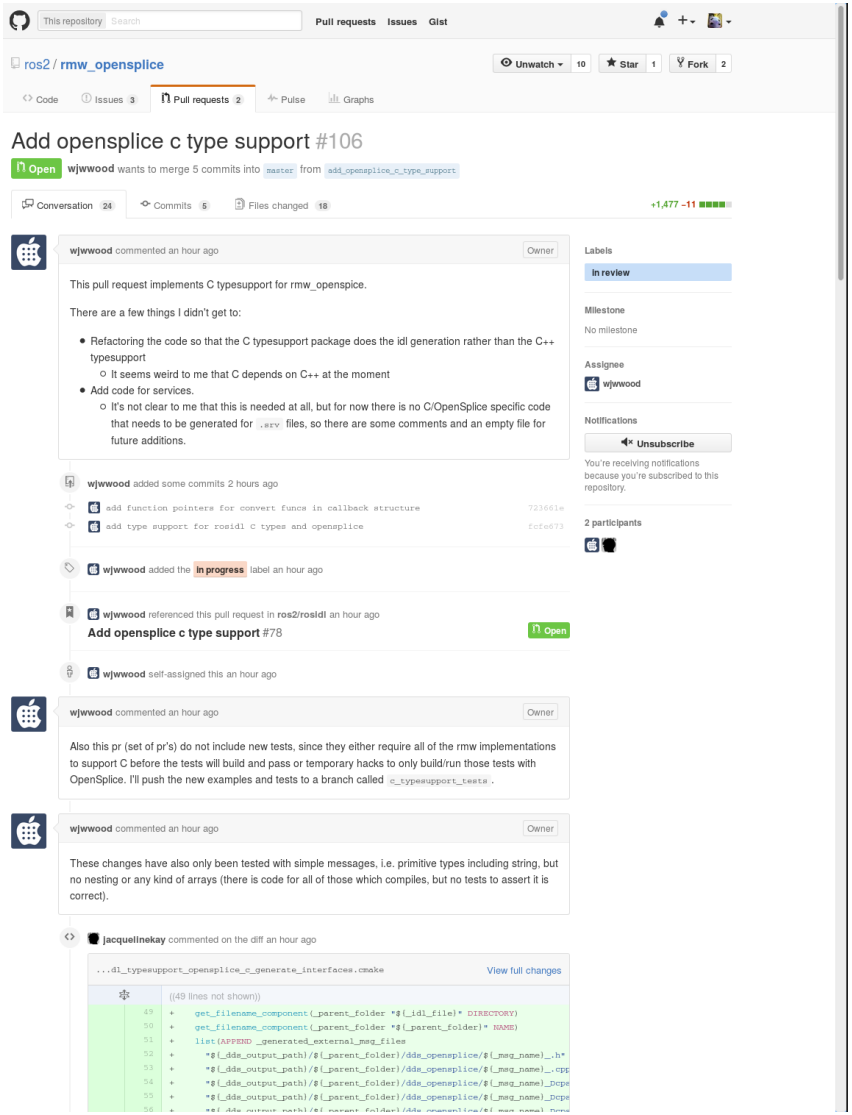


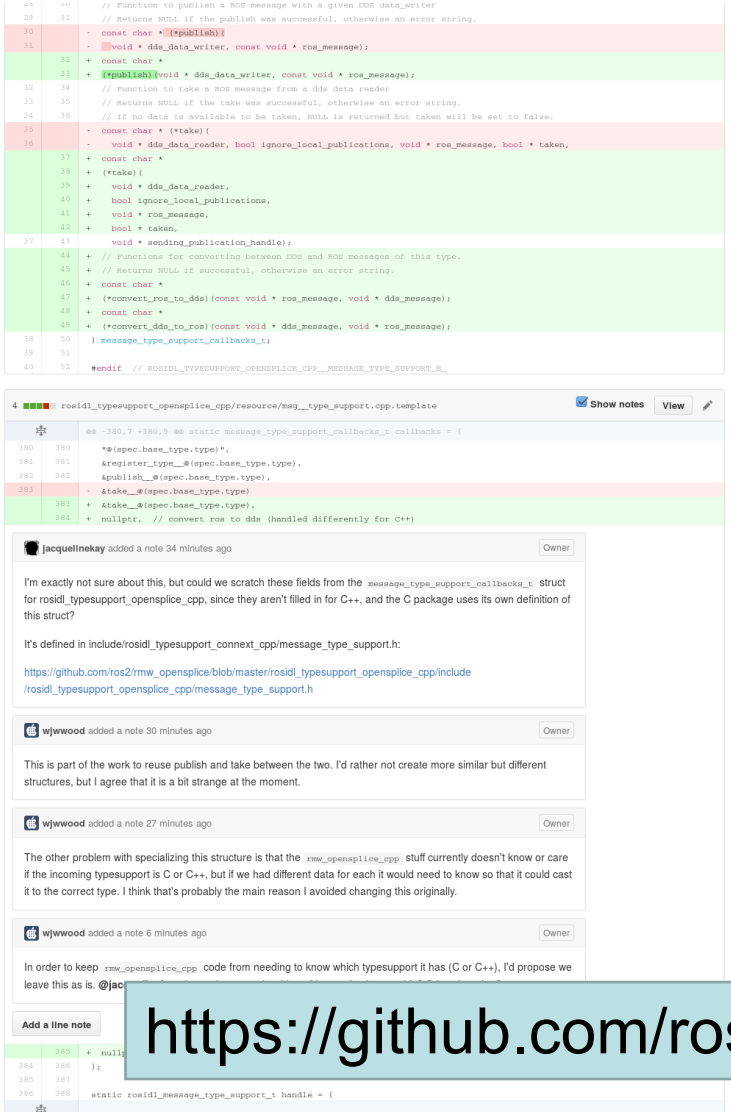
オープンソース、オープン開発



https://waffle.io/ros2/ros2

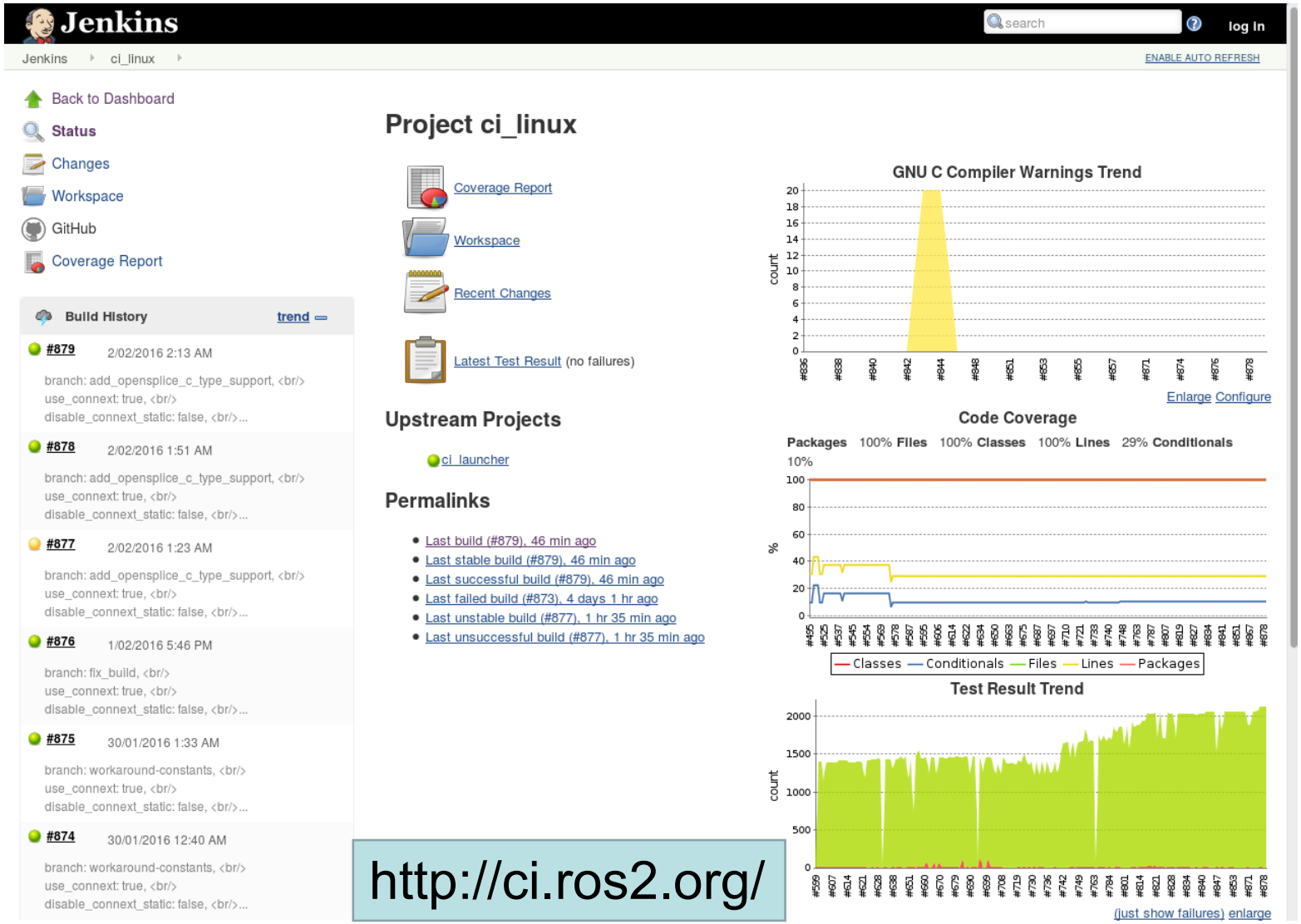
オープンソース、オープン開発





https://github.com/ros2

オープンソース、オープン開発



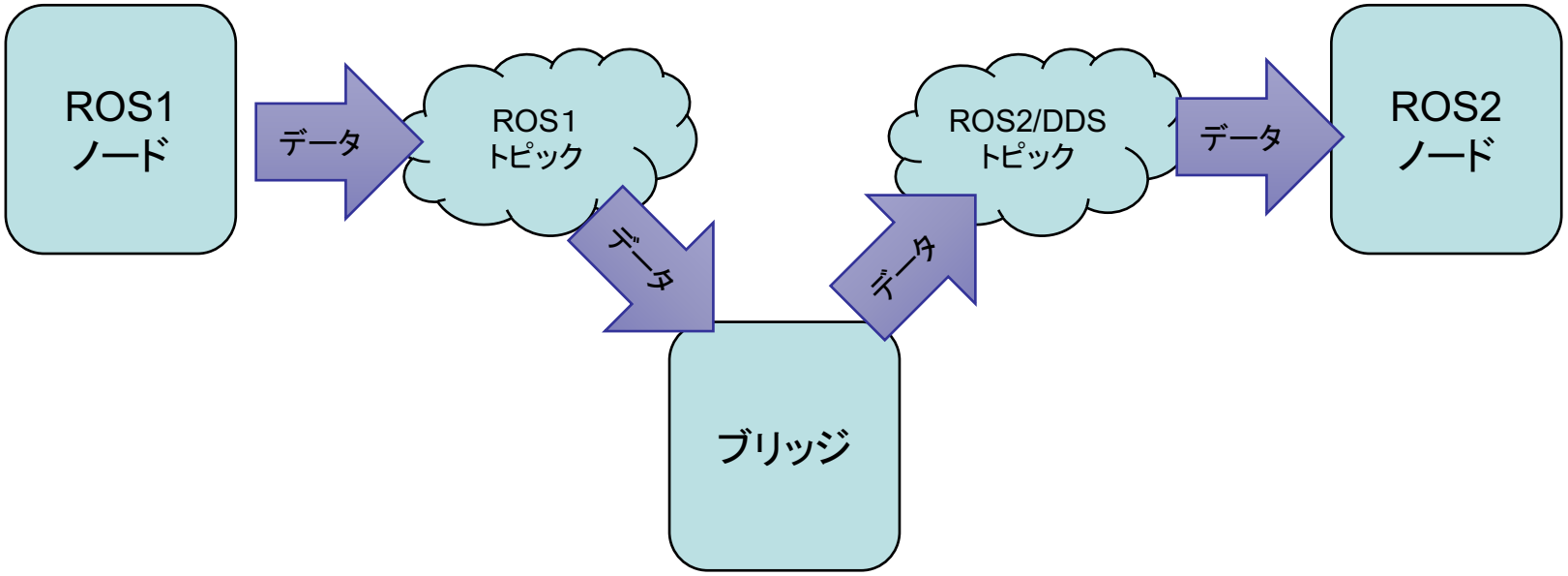
http://ci.ros2.org/

認証可能なROSへ

- 開発プロセスが追跡できたから認証可能
- アーキテクチャーは認証を支援
  - 必要な部分のみ利用し、それを認証する
- MISRA-Cに合致
- 静的システム作成可能
  - 完全静的ROS2は開発中（Apex.io）
- セキュリティは基礎からサポートあり
  - DDS-Security規格に合致

ROS1 と インタラクション

- ブリッジによりROS1のシステムとROS2のシステムが通信可能



ROS2のまとめ

- 企業世界で好評された通信ミドルウェアの採択
- ノードをより構造化に
  - 決定的なロボットが作成可能
  - リアルタイム対応可能
- Windows、組み込みシステムがサポートされたプラットフォームに
- 企業のニーズに合うロボットソフトウェアプラットフォーム

Part 5

これからのROS 2

現在状況

- まだフルスピードで開発中
- 利用が加速中
  - ROSConでROS 2についての発表は増えている
- リリースサイクルは半年
  - 次のリリースは6月・7月
- まだ開発されていないROS 1のフィーチャーあり
  - actionlib
  - MoveIt!
  - …

ROS Ardent Apalone（2017/12）

- 最初のサポートするリリース
  - バグフィクス等をリリースする
- 最初の「利用するべき」なリリース
  - プレビューではない
- 全パッケージはバージョン1ではない！
  - rmw: 0.4.0　←一番中心のライブラリ
  - rclcpp: 0.4.0
  - tf2_eigen: 0.8.0
  - rviz2: 2.0.0

ROS Bouncy Bolson（2018/06）

- 2回目のサポートするリリース
  - プレビューではない
- パッケージのバージョンアップもあり
- 新フィーチャもあり
  - 新しいlaunchツール
  - ライフサイクル管理ツール
  - パラメータサポート拡張
  - Rvizの機能大拡張
  - など

実装予定のフィーチャー

- アクションの実装
- の完成
- メッセージIDLの改善
- 動的なPythonベースlaunchツール
- ネイティブなrosbag
- リアルタイムサポートの拡張
- セキュリティサポートの拡張

https://github.com/ros2/ros2/wiki/Roadmap

ROS 2の開発に参加しよう

- デザイン情報を読む
  - コメントを提供
  - 改善する（プルリクエスト）
  - 書く！（新しいデザインドキュメントを提供する）
  - 自分の会社のニーズを伝える時は今だ！

http://design.ros2.org/

ROS 2の開発に参加しよう

- コードレビューに参加
  - ROS 2のコアライブラリの実装はすべて
    コードレビュー経由で開発されている
  - 例：https://github.com/ros2/rclcpp/pulls

https://github.com/ros2

ROS 2の開発に参加しよう

- コードを書く
  - ROS 2のカンバンから小さなタスクを選択する
  - タスクを行う
  - プルリクエストを送る！

https://waffle.io/ros2/ros2

https://gbiggs.github.io/rosjp_ros2_intro/