SONY

aiboにおけるROSの利用とソニーの取り組み

ソニー株式会社
R&Dセンター
藤田智哉

R&D Center System Technology Development Division Base System Development Department

Agenda

- 背景
- ROS エンベデッド最適化例
  - Unix Domain Socket
  - Direct I/O
  - EPOLL
  - Polling Frequency
- ROS2に向けて
- ソニー ロボティクスソフトウェア

背景

- ROS
  - オープンソース
  - 豊富なパッケージ
  - シミュレーション／デバッグツール
  - ビルドツール
  - メッセージ通信
  - デバイスドライバ
- 組み込み課題
  - リアルタイム
  - リソース消費

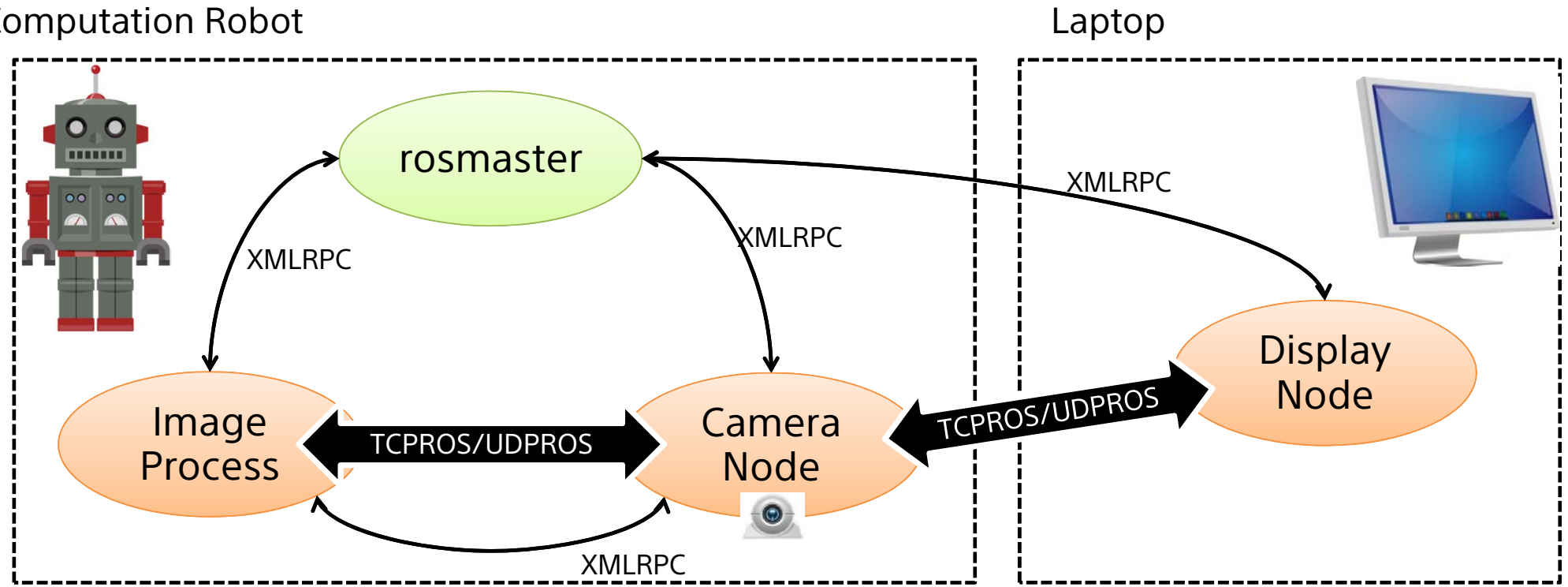
組み込み製品にROSを搭載するために行った最適化や施策を紹介

ROSエンベデッド最適化事例

ROS Transport Overview

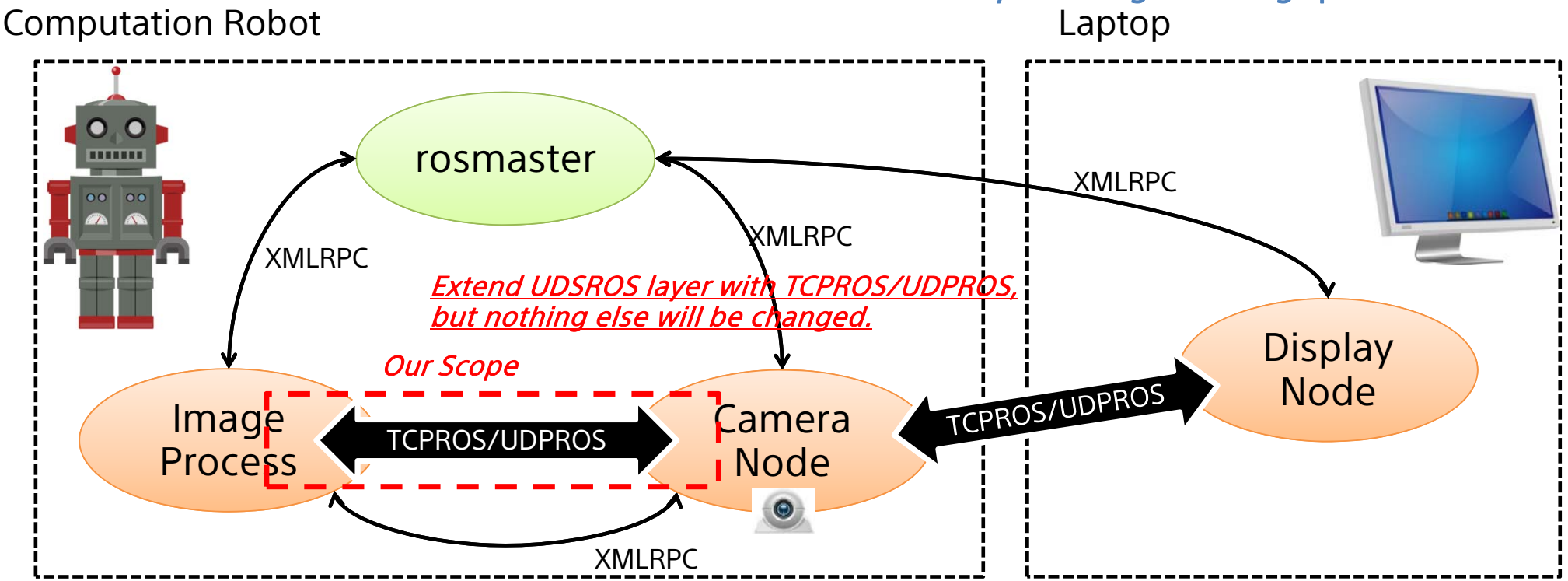
- ROS Concept
  - TCPROS(TCP/IP) and UDPROS(UDP/IP).
  - IP is world wide experience via ether.

XMLRPC:
configuration information
TCPROS/UDPROS:
data payload such as topic and services.



ROS Transport Overview

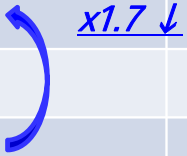
- What's the problem?
  - CPU resource stress
  - Software overhead for IP protocol.
- Countermeasure Preconditions
  - No change required for applications.
  - Good affinity with TCP/UDP
  - small latency and high throughput.

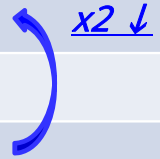


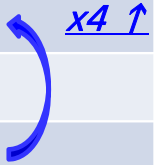
Expected Improvement as System

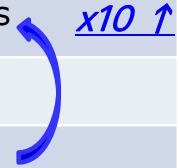
- Source Code
  - https://github.com/tomoyafujita/ipc-bench
- Environment
  - skylake(amd64) / hikey(arm64)
  - CPU frequency governor is “performance”
  - 100byte data payload / 10000 test loop average
- Performance Improvement for System

Category		Skylake	Hikey
Latency	UDS(STREAM)	1.82 us	44.4 us
[usec]	UDS(DGRAM)	2.14 us	19.9 us
	TCP/IP	3.14 us	87.7 us
	UDP/IP	2.78 us	67.2 us
Through-Put	UDS(STREAM)	1.95 Gbps	127 Mbps
	UDS(DGRAM)	1.98 Gbps	92 Mbps
	TCP/IP	0.43 Gbps	12 Mbps

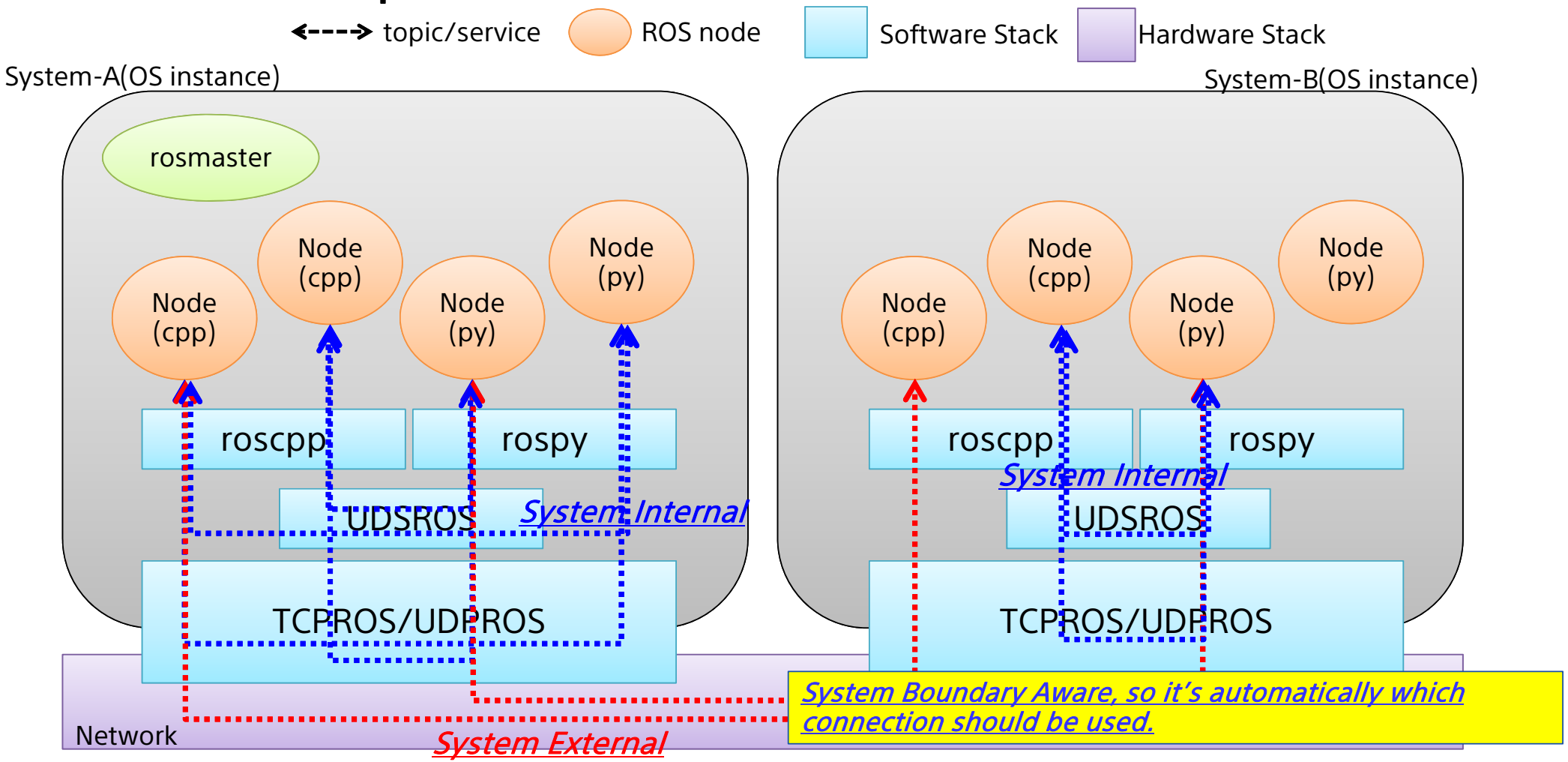








Extended Transport Overview



Improvement Result

- HelloWorld Benchmark
  - Publisher:Subscriber=1:1
  - Synchronous test loop using shared memory.
  - Latency is the time window between publish and callback entrance.
  - Environment: amd64(skylake) and arm64(hikey)

- Result Comparison

Platform	ROS Transport	Duration [nsec]		
		min	max	average
Skylake	TCP/IP	46,582	287,173	70,393
	UDS	42,877	286,130	64,796
Hikey	TCP/IP	552,224	6,040,312	1,145,881
	UDS	435,833	10,075,833	872,981

8% ↓

24% ↓

Source Code

Github URL

<will be released soon>

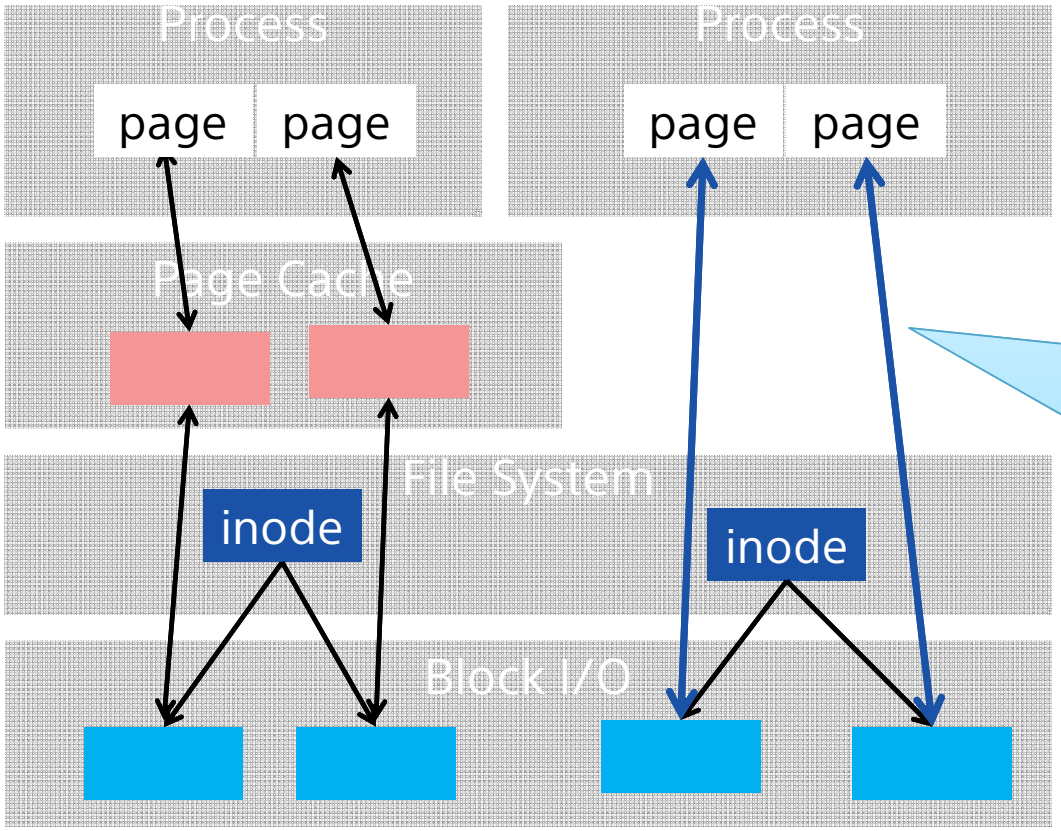
Maintainer

Tomoya, Fujita <Tomoya.Fujita@sony.com>

Xu, Barry <Barry.Xu@sony.com>

Direct I/O with rosbag

- Reference
  - https://github.com/osrf/rosbag_direct_write



Data in userland will be stored directly to the storage device without CPU workload.

(*) memory alignment constraints

Epoll

- What is the problem?
  - The more ROS topic connection, the more CPU stress with ros::spin()
    - Waiting for the stimulus during ros::spin without any message transmission.
  - Poll system call is not good enough to take care of many file descriptors.

- Countermeasure
  - Using epoll instead, to reduce CPU stress.
  - Already introduces as following commit, so backport the fix into 1.12.7 ros_comm.

```
commit 9c0db37d9231003a7f162857e4aeb45675839609
Author: Mike Purvis <mike@uwmike.com>
Date:   Thu Dec 21 11:31:08 2017 -0500

    Topic subscription scalability fix (#1217)

    Switching to using epoll system calls to improve performance of the topic polling code by a factor of 2.
    This required disabling the addDelMultiThread test.
```

Polling Frequency

- ROS Threads

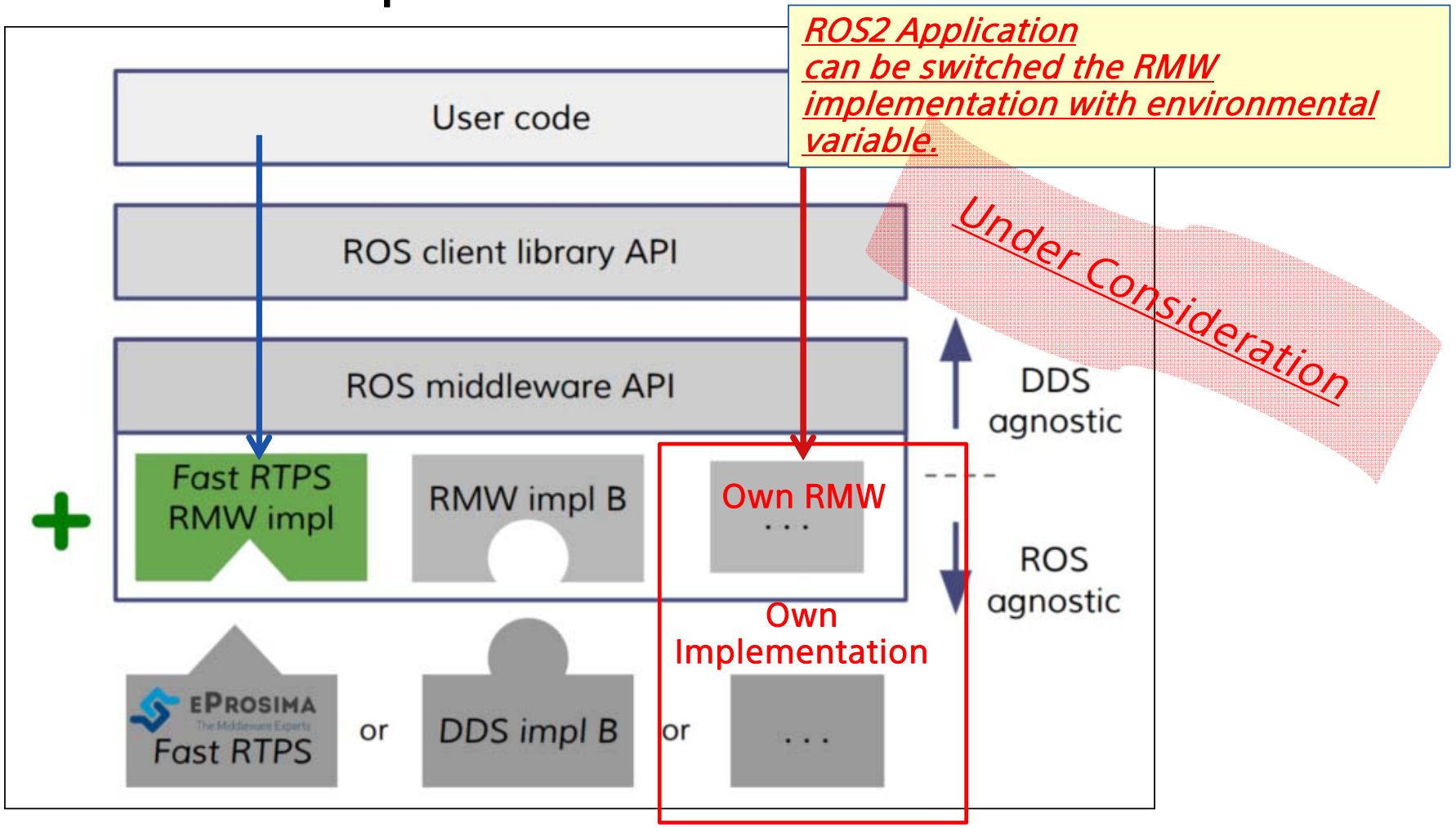
Thread Name	Use
Main	Main thread
PollManager	Publish/Subscribe I/O Polling
XMLRPCManager	Talk to rosmaster for connectivity map and services
ROSOutAppender	Logging
internalCallbackQueue	Connection drop, data store etc…

Polling Frequency Tunable for each processes, less CPU stress. (default freq is 100msec)

No Logging thread, less CPU stress. (instead of that, using own log system)

ROS2

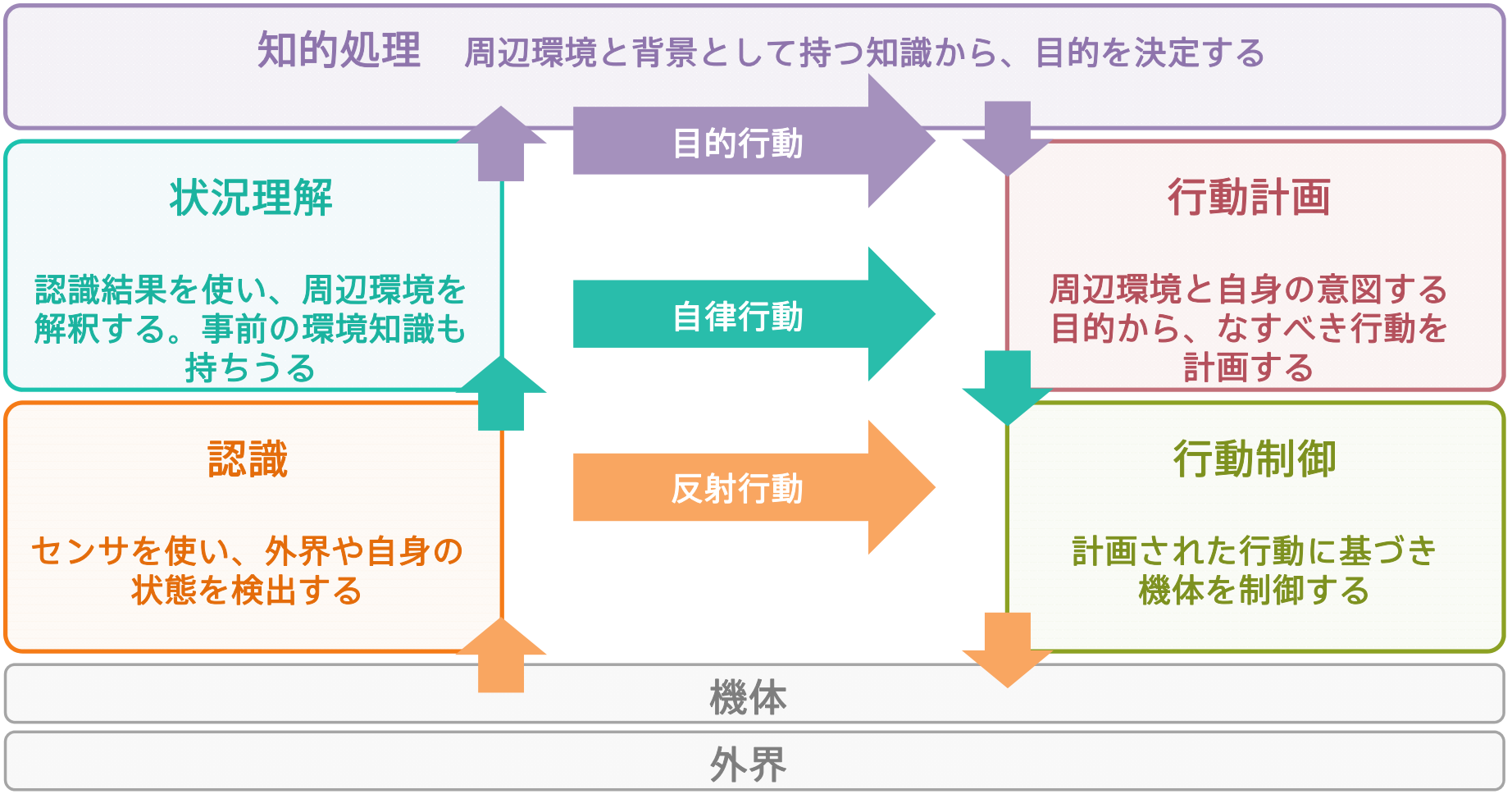
ROS2 RMW with Own Implementation



https://roscon.ros.org/2016/presentations/ROSCon%202016%20-%20ROS%202%20Update.pdf

ソニー ロボティクスソフトウェア

Architecture Overview

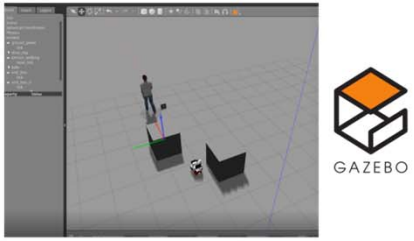


Componential Overview

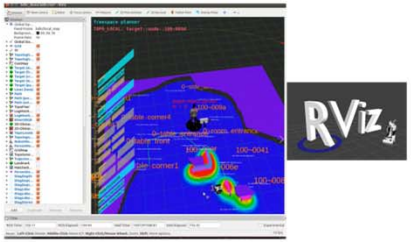
Development Environment

シミュレータを活用した開発
Extensive use of simulation

物理演算シミュレータ
Gazebo simulator / Physics Engine

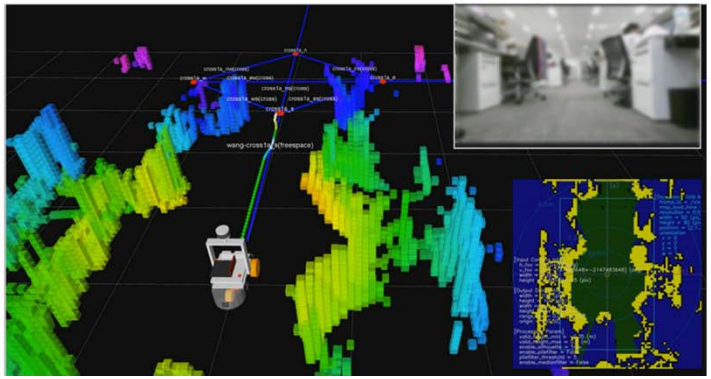


可視化ツール
Rviz – ROS visualizer



Sim to real, real to sim

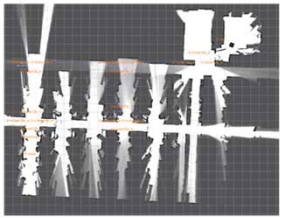
Testbedを用いて実機検証しながら開発
SW development using testbed, in real world



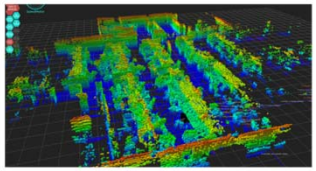
Thanks to ROS

ROSエコシステムのOSS群の活用により開発加速
Accelerate SW development using ROS eco-system

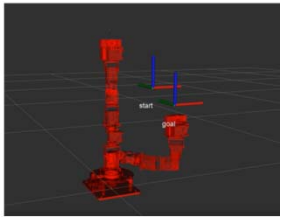
Cartographer-ros



Octomap



MoveIt!



Let's make a difference with Sony

SONY