International Telecommunication Union
Standardization Sector

Recommendation

ITU-T Y.3533 (12/2023)

SERIES Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities

Cloud Computing

Cloud computing – Functional requirements for robotics as a service

ITU-T Y-SERIES RECOMMENDATIONS

Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities

GLOBAL INFORMATION INFRASTRUCTURE	Y.100-Y.999
General	Y.100-Y.199
Services, applications and middleware	Y.200-Y.299
Network aspects	Y.300-Y.399
Interfaces and protocols	Y.400-Y.499
Numbering, addressing and naming	Y.500-Y.599
Operation, administration and maintenance	Y.600-Y.699
Security	Y.700-Y.799
Performances	Y.800-Y.899
INTERNET PROTOCOL ASPECTS	Y.1000-Y.1999
General	Y.1000-Y.1099
Services and applications	Y.1100-Y.1199
Architecture, access, network capabilities and resource management	Y.1200-Y.1299
Transport	Y.1300-Y.1399
Interworking	Y.1400-Y.1499
Quality of service and network performance	Y.1500-Y.1599
Signalling	Y.1600-Y.1699
Operation, administration and maintenance	Y.1700-Y.1799
Charging	Y.1800-Y.1899
IPTV over NGN	Y.1900-Y.1999
NEXT GENERATION NETWORKS	Y.2000-Y.2999
Frameworks and functional architecture models	Y.2000-Y.2099
Quality of Service and performance	Y.2100-Y.2199
Service aspects: Service capabilities and service architecture	Y.2200-Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250-Y.2299
Enhancements to NGN	Y.2300-Y.2399
Network management	Y.2400-Y.2499
Computing power networks	Y.2500-Y.2599
Packet-based Networks	Y.2600-Y.2699
Security	Y.2700-Y.2799
Generalized mobility	Y.2800-Y.2899
Carrier grade open environment	Y.2900-Y.2999
FUTURE NETWORKS	Y.3000-Y.3499
CLOUD COMPUTING	**Y.3500-Y.3599**
BIG DATA	Y.3600-Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800-Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	Y.4000-Y.4999
General	Y.4000-Y.4049
Definitions and terminologies	Y.4050-Y.4099
Requirements and use cases	Y.4100-Y.4249
Infrastructure, connectivity and networks	Y.4250-Y.4399
Frameworks, architectures and protocols	Y.4400-Y.4549
Services, applications, computation and data processing	Y.4550-Y.4699
Management, control and performance	Y.4700-Y.4799
Identification and security	Y.4800-Y.4899
Evaluation and assessment	Y.4900-Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3533

Cloud computing – Functional requirements for robotics as a service

Summary

Robotics as a service (RaaS) is a cloud service category aimed at supporting the development, operation and management of robots in a cloud computing environment. Recommendation ITU-T Y.3533 provides an overview of robotics, robotics as a service, and functional requirements through various use cases of robotics in cloud computing. Also, this Recommendation is aligned with the cloud computing reference architecture of Recommendation ITU-T Y.3502.

History *

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T Y.3533	2023-12-14	13	11.1002/1000/15746

Keywords

Cloud computing, RaaS, robotics as a service.

* To access the Recommendation, type the URL https://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
3.1	Terms defined elsewhere	1
3.2	Terms defined in this Recommendation	2
4	Abbreviations and acronyms	2
5	Conventions	3
6	Overview of robotics	3
6.1	Introduction to robotics	3
6.2	Generic robotics life cycle	4
6.3	Robotics ecosystem	6
7	Overview of robotics as a service	8
7.1	Introduction to RaaS	8
7.2	Logical components of RaaS	10
7.3	System context of RaaS	11
7.4	CSN:robotic module developer	12
7.5	CSN:robotic platform developer	13
7.6	CSP:robotics service provider	14
7.7	CSC:robotics service customer	15
8	Functional requirements of cloud computing for RaaS	16
8.1	Robotic module support	16
8.2	Robotic platform support	17
8.3	Robot deployment support	17
8.4	Robot operation support	18
9	Security considerations	19
Appendix I – Use case of robotics as a service		20
I.1	Registration of robotic module profiles and development of robot applications using the profiles	20
I.2	Operation of robot intelligence using RaaS in the environment of an industrial roller conveyor with multi-gripper	21
I.3	Enhancement of environment perception of a robot using RaaS	22
I.4	Developing and testing a robot system with a fine-tuning navigation module	23
I.5	Deployment of the distributed robots using RaaS & cloud service	24
I.6	Extension of robot system with adding robotic modules	26
I.7	Fine-tuning of robotic modules with recalibration in RaaS service	27
I.8	Remote control and offloading robots in a cloud environment via network	28
Appendix II – Use case of robotics as a service for application perspective		30

Page

II.1 Operations of robots with cloud and network connectivity in warehouse environment 30
II.2 Medical robot software development on the cloud environment 31
II.3 Operating agriculture robot system with cloud management 32
Appendix III – Technical Concepts of modularity, virtualized simulation environment and robotic system deployment in robotics 34
III.1 Robotic modules and modularity in robotics 34
III.2 Virtual robot representation 35
III.3 Robot system deployment: robot-server architecture 36
Bibliography 38

Recommendation ITU-T Y.3533

Cloud computing – Functional requirements for robotics as a service

1 Scope

This Recommendation provides an overview and the functional requirements for robotics as a service (RaaS) in the cloud environment. This Recommendation addresses the following subjects:

− Concept and overview of robotics;

− Introduction and system context of robotics as a service;

− Functional requirements for robotics as a service;

− Use cases of robotics as a service.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3502] Recommendation ITU-T Y.3502 (2014), *Information technology – Cloud computing – Reference architecture*.

[ITU-T Y.3535] Recommendation ITU-T Y.3535 (2022), *Cloud computing - Functional requirements for a container*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 activity [ITU-T Y.3502]: A specified pursuit or set of tasks.

3.1.2 autonomy [b-ISO 8373]: Ability to perform intended tasks based on current state and sensing, without human intervention.

3.1.3 cloud computing [b-ITU-T Y.3500]: Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.

NOTE – Examples of resources include servers, operating systems, networks, software, applications and storage equipment.

3.1.4 cloud service [b-ITU-T Y.3500]: One or more capabilities offered via cloud computing invoked using a defined interface.

3.1.5 cloud service customer [b-ITU-T Y.3500]: Party which is in a business relationship for the purpose of using cloud services.

3.1.6 cloud service partner [b-ITU-T Y.3500]: Party which is engaged in support of, or auxiliary to, activities of either the cloud service provider or the cloud service customer, or both.

3.1.7 cloud service provider [b-ITU-T Y.3500]: Party which makes cloud services available.

3.1.8 robot [b-ISO 8373]: Programmed actuated mechanism with a degree of autonomy to perform locomotion, manipulation or positioning.

3.1.9 robotics [b-ISO 8373]: Science and practice of designing, manufacturing, and applying robots.

3.1.10 role [ITU-T Y.3502]: A set of activities that serves a common purpose.

3.1.11 sub-role [ITU-T Y.3502]: A subset of the activities of a given role.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 robotic module: A set of abstracted hardware and software that is part of a robot (3.1.8).

NOTE – The abstracted hardware refers to the software representation of hardware aspect in robot, which is designed for simulations and operations.

3.2.2 robotic platform: A set of tools and libraries to support robotics (3.1.9).

NOTE 1 – Tools include an integrated development environment (IDE), a robot simulator, etc.

NOTE 2 – Libraries include a packaging robotic module, a planning actions for a robotic task, middleware for robot operation, etc.

3.2.3 robotic task: A set of actions that a robot (3.1.8) performs, which is designed to achieve a goal in a certain scenario or environment.

3.2.4 robotics as a service: A cloud service category in which the capabilities provided to a cloud service customer is the provision of functionalities for robotics (3.1.9).

NOTE – Robotics as a service (RaaS) provides the functionalities to design, build, simulate, deploy, operate, and manage a robot (3.1.8).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

CSC Cloud Service Customer

CSN Cloud Service Partner

CSP Cloud Service Provider

HW Hardware

IDE Integrated Development Environment

OGRE Object-Oriented Graphics Rendering Engine

OTA Over-The-Air

RaaS Robotics as a Service

RMD Robotic Module Developer

RPD Robotic Platform Developer

RSC Robotics Service Customer

RSP Robotics Service Provider

SDF Standard Digital Format

SW Software

URDF Unified Robot Description Format

5 Conventions

The following conventions are used in this Recommendation:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Overview of robotics

6.1 Introduction to robotics

Robotics (3.1.9) is a science and a practice of designing, manufacturing, and applying robots (3.1.8). The scope of robotics encompasses various technical domains such as engineering, mechanical, electronic and software technologies.

The primary objective of robotics is to develop robots to perform a robotic task (3.2.3) to assist and aid human beings. A robotic task is broken down into a series of smaller actions that are designed and managed by a robotics engineer.

NOTE 1 – An example of a robotic task is 'pick an object with grasper' task. The robotic task is split into a series of actions including 'planning grasp approach', 'moving a grasper to target object', 'controlling grasping power ', etc.

A robot is composed of robotic modules (3.2.1) to perform required robotic tasks. In a single robot, each robotic module performs specific actions such as controlling a motor, interpreting sensor data, performing calculations of power degree, etc.

A robotic platform (3.2.2) is used to design robotic tasks and develop robotic modules for robots. A robotic platform is a collection of tools and libraries such as packaging software and hardware components into a robotic module, or integrating the robotic modules for a single robot. The robotic platform not only supports development of a robot, but also supports operation and management of a robot.

Figure 6-1 shows a conceptual representation of a robot, a robotic task, a robotic module and a robotic platform including interrelationship among robotics concepts.

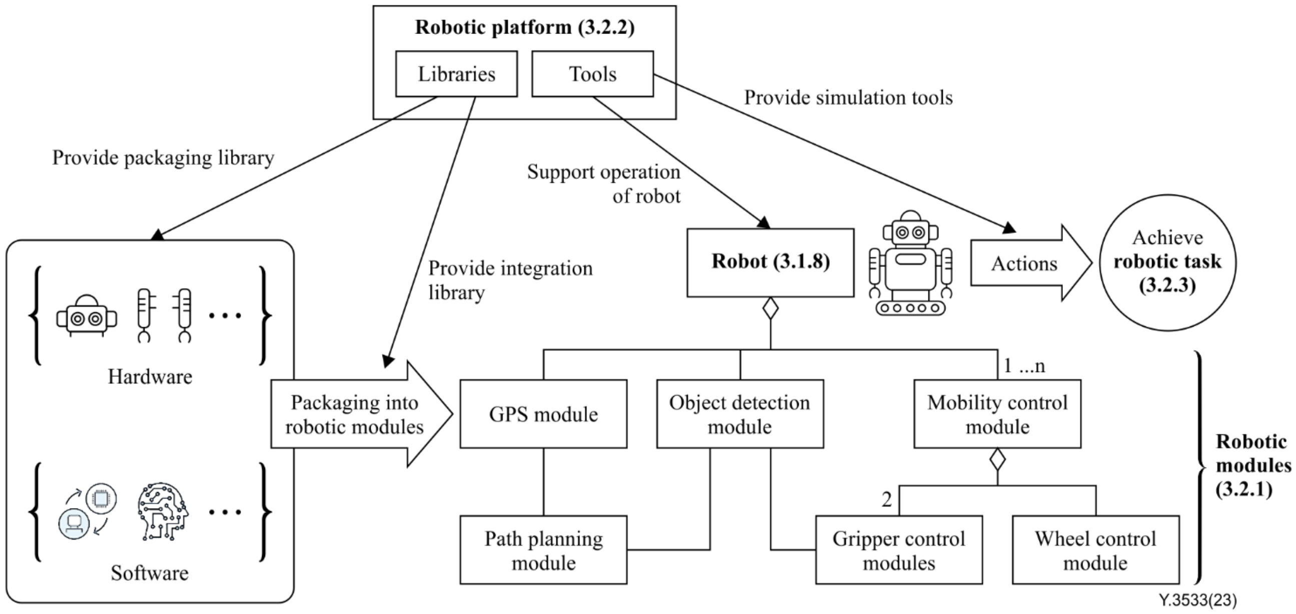


Figure 6-1 – Conceptual representation of a robot, a robotic task, a robotic module, and a robotic platform

As Figure 6-1 represents, hardware and software components are packaged into robotic modules when developing the robots. Packaging robotic modules involve adding interfaces to control hardware and software components as a single robotic module to be operated in the robot, as well as making the corresponding profile for a robotic module.

NOTE 2 – This Recommendation does not consider the classification of robotic module, which is described in [b-ISO 22166-1].

NOTE 3 – Profile is information that provides the attributes of a robotic module to support robot design, interoperability and re-use.

Appendix III provides more information and examples of robotics.

6.2 Generic robotics life cycle

Robotics includes the practice of designing, manufacturing, and applying robots. Figure 6-2 illustrates a generic robotics life cycle including detailed activities at each of the designing, manufacturing and applying stages. Validation and verification are performed accordingly at each stage, while maintenance and management affect the manufacturing and applying stages. The detailed activities are as follows:

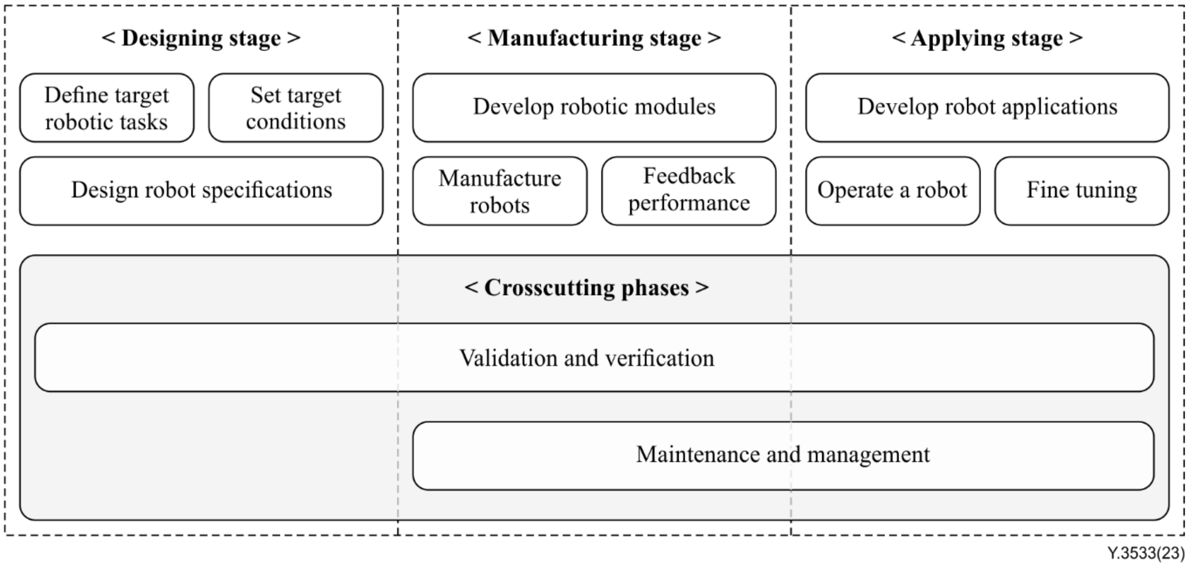


Figure 6-2 – Generic life cycle for robotics

The designing stage includes:

– **Define robotic task**: Determines the specific robotic tasks that the robot is intended to perform and expected outcomes, considering functional requirements, performance criteria and the complexity of the robotic tasks.

– **Set target condition**: Identifies the environments of robot operation such as industrial and home environment. Derive the specific constraints of the environment including place of use, climate conditions, robot-human interaction, etc.

– **Design robot specifications**: Derives the robot specification including the robot's physical structure and software mechanisms. This includes the virtual test of the robot in the target condition to meet the requirements for the defined robotic task.

The manufacturing stage includes:

– **Develop robotic modules**: Implements robot specifications with module-level designing. This phase verifies the robot's physical specification with embedded software and software parts and verifies if it performs or operates correctly as intended in the designing phase.

– **Manufacture robots**: Integrates the developed robotic modules with connecting interfaces in robots. During this phase, a system to control the robot is developed.

  NOTE 1 – This phase defines the system commands to control a single robot to operate robotic tasks. A user-level guide for commands and interfaces is developed.

– **Feedback performance**: Involves the iterative analysis process of the outcomes of manufactured robots. The observed issues from the expected results are documented, and feedback is provided to the stakeholders. This ensures that necessary adjustments are made, promoting the continuous improvement of the robotic system.

The applying stage includes:

– **Develop robot application**: Creates specific applications for the robot to perform expected robotic tasks. Depending on the robot's specifications, applications range from simple robotic tasks to complex operations.

– **Operate a robot**: Deploys robots into the intended environment. Robots begin performing the tasks in the real world. The operation phase also involves monitoring the robot's performance, and ensuring that it operates optimally and safely in its working conditions.

– **Fine tuning**: Makes subtle adjustments to the robot's performance based on real-world feedback and operational data after deployment. This phase focuses on the adjustments, optimizing the robot's functionalities to suit its environment or improve its efficiency. This involves software updates, recalibrations and minor hardware modifications.

Crosscutting phases during generic life-cycle includes:

– **Validation and verification**: This crosscutting phase influences all the stages. Validation ensures that developing robot or robotic modules meets users' needs and performs appropriately in the given conditions, while verification corroborates the implementation of the robot's functions and specifications. For example, verification checks if the robot's sensors detect obstacles as intended, whereas validation tests if the robot navigates effectively around real household objects.

  NOTE 2 – Specific methods for validation and verification are not the scope of this Recommendation.

– **Maintenance and management**: This is performed after the manufacturing stage to support robot developments. This phase includes updating robotic modules, upgrading the hardware for robotic modules, and repairing robots. Specific methods for maintenance and management are outside the scope of this Recommendation.

6.3 Robotics ecosystem

This clause describes a robotics ecosystem through roles and sub-roles or stakeholder's related with robotics. It provides activities of the roles to provide and consume robotics as well as the relationships between roles.

The robotics ecosystem includes following roles:

– Robotic module developer;

– Robotic platform developer;

– Robot provider;

– Robot customer.

Figure 6-3 shows the robotics ecosystem.

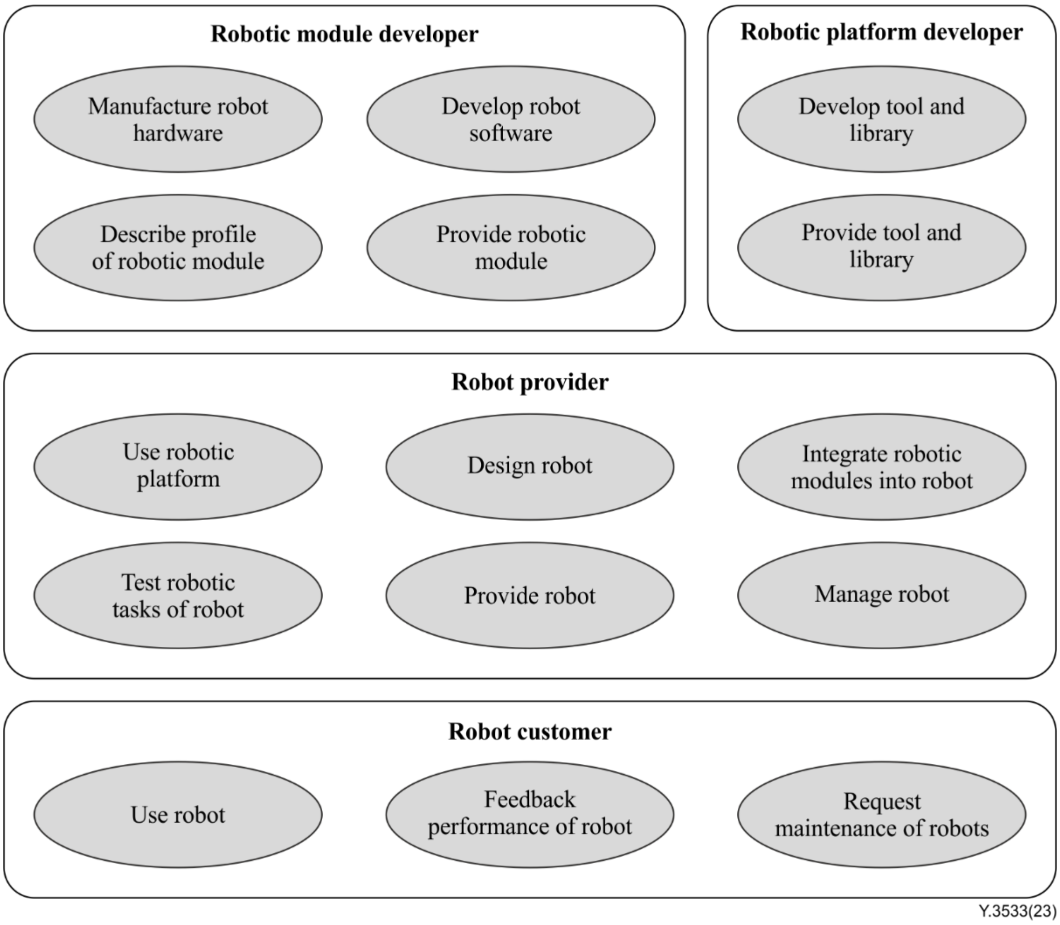


Figure 6-3 – Robotics ecosystem

6.3.1 Robotic module developer

A robotic module developer involves developing robotic modules and profiles for a robotic module repository.

The robotic module developer's activities include:

– Manufacture robot hardware;

NOTE – The robot hardware includes both mechanical parts of robots and software for control of hardware such as sensors, skeletons, cameras, motors, etc.

– develop robot software;
– describe the profile of robotic modules;
– provide robotic modules.

6.3.2 Robotic platform developer

A robotic platform developer provides a robotic platform which contains tools and libraries for supporting robot development.

The robotic platform developer's activities include:

– Development of a tool and a library for robotics;
– provision of a tool and a library for robotics.

6.3.3 Robot provider

A robot provider provides the integration of robotic modules in the robotic platform.

The robot provider's activities include:

– Use a robotic platform;
– Design a robot;
– Integrate robotic modules into a robot;
– Test a robotic task of a robot;
– Provide a robot;
– Manage a robot.

6.3.4 Robot customer

A robot customer uses a robot that has been developed by a robot provider. The robot customer is an end-user of a robotics ecosystem who purchases and uses robots.

The robot customer's activities include:

– Using a robot;
– Providing feedback of the performance of a robot;
– Requesting maintenance of a robot.

7 Overview of robotics as a service

7.1 Introduction to RaaS

Robotics as a service (RaaS) is a cloud service category in which the capabilities provided to a cloud service customer is the provision of functionalities for robotics. To support RaaS service, cloud service provider (CSP) provides the functionalities for supporting robotics activities as described in clause 6.3 during generic life-cycle of robotics including designing, manufacturing, and applying stages using cloud resources. Figure 7-1 illustrates an example of utilizing RaaS functionalities which supports to simulate the robotic modules to execute robotic tasks in a cloud environment.

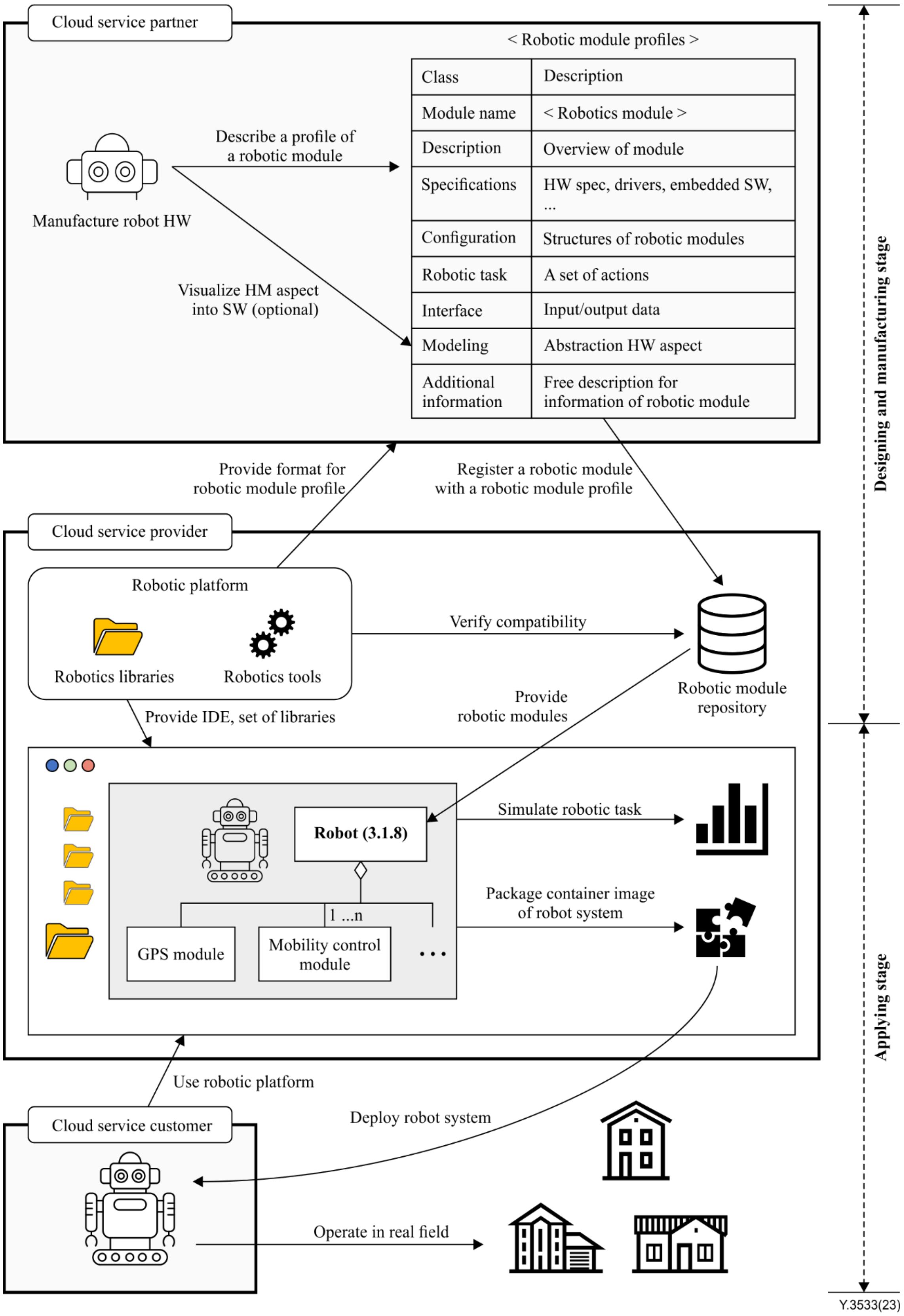


Figure 7-1 – An example of robotics as a service and relationships with generic life cycle

Figure 7-1 includes the relationships between cloud service and generic life cycle of robotics. All phases of designing and manufacturing stages are involved with the activities of CSP and cloud service partner (CSN) and cloud service customer (CSC). The CSN is responsible for the development and manufacturing of robotic modules including hardware aspects of the modules. The CSP provide the RaaS service to develop a robot system for a robot. The robot system is a software on the cloud that integrates robotic modules including the control interface of the robot. The CSP support container engines to package a robot system into a container image. The robot system is managed as a deployable format, and executed in the robot.

NOTE 1 – The functional requirements of a container are described in [ITU-T Y.3535].

The CSC participates in the applying stage for developing and simulating robotic tasks for robots and requesting the support of cloud for operating robots. The support for operating robots includes the fine tuning for optimizing the robot's capabilities and maintenance to manage robots efficiently with cloud services.

NOTE 2 – Providing robot hardware to a CSC is outside the scope of RaaS.

7.2 Logical components of RaaS

This clause describes the logical components of RaaS for supporting robotics in a cloud environment. The logical components are introduced as the conceptual building blocks of a RaaS. Figure 7-2 illustrates the logical components of RaaS.

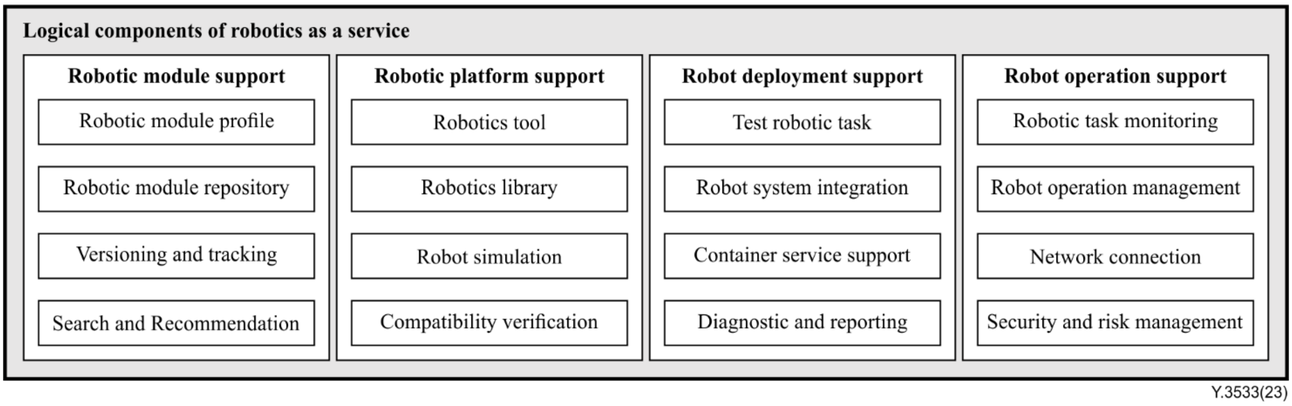


Figure 7-2 – Logical components of RaaS

The logical components shown in Figure 7-2 are as follows:

– **Robotic module support**: This logical component manages the robotic module to provide validated robotic modules through the cloud service. The main feature of this logical component includes configuring common robotic module profile, and maintaining the robotic module repository. Versioning and tracking robotic modules to control unexpected issues, retrieve research functionalities and provide advice is under the responsibility of this logical component.

– **Robotic platform support**: This logical component provides the tools and libraries for robotics as an integrated platform to support robot development. Compatibilities among the tools, libraries and robotic modules are verified with this component. The simulation of robotic tasks within controlled conditions are provided, and virtual robot representations are optionally supported. User-defined configurations and settings are available including simultaneous simulations of multiple robots.

  NOTE – The virtual robot representation is an abstraction of the movements of the robot's hardware aspect.

– **Robot deployment support**: This logical component supports transformation of a robot system into deployable formats and their integration onto physical robots. The real field test suites for robotic tasks and the container service for the robot system are provided. The malfunctions and unexpected issues encountered during the deployment are reported.

– **Robot operation support**: This logical component provides functionalities of operations, remote control and monitoring robots through the cloud environment. It gathers data from the robot operations to provide performance metrics. Tools for consequent troubleshooting and debugging utilities for robots are provided. Robust network connections between cloud and robot are supported with fail-safe protocols. Security and risk management are supported with awareness of limitations of cloud capabilities.

7.3 System context of RaaS

This clause describes a cloud computing-based system context for RaaS that is effective for supporting cloud services to support robotics.

System context of RaaS provides additional sub-roles and activities based on the architectural user view defined in [ITU-T Y.3502]. This clause describes how cloud computing supports the four main roles in a RaaS ecosystem: robot hardware provider, robot software provider, robot provider, and robot customer.

Cloud computing sub-roles can be mapped to industrial robotics ecosystem roles and activities as shown in Table 7-1. The sub-roles of CSP, CSN, and CSC are mapped to the roles and sub-roles of robotics service ecosystems.

Table 7-1 – Mapping table between robotics service ecosystem and RaaS system context

Robotics ecosystem (clause 6.3)		**RaaS System context (clauses 7.4, 7.5, 7.6, and 7.7)**
Role	**Activity**	
Robotic module developer	Manufacture a robot hardware	CSN: RMD (Robotic module developer)
	Develop a robot software	CSN: RMD (Robotic module developer)
	Describe a profile of a robotic module	CSN: RMD (Robotic module developer)
	Provide a robotic module	CSP: RSP (Robotics service provider)
Robotic platform developer	Develop a tool and a library for robotics	CSN: RPD (Robotic platform developer)
	Provide a tool and a library for robotics	CSP: RSP (Robotics service provider)
Robot provider	Use a robotic platform	CSC: RSC (Robotics service customer)
	Design a robot	CSC: RSC (Robotics service customer)
	Integrate robotic modules into a robot	CSP: RSP (Robotics service provider)
	Test a robotic task of a robot	CSP: RSP (Robotics service provider)
	Manage a robot	CSP: RSP (Robotics service provider)
Robot customer	Use a robot	CSC: RSC (Robotics service customer)

Table 7-1 – Mapping table between robotics service ecosystem and RaaS system context

Robotics ecosystem (clause 6.3)		RaaS System context (clauses 7.4, 7.5, 7.6, and 7.7)
Role	**Activity**	
	Feedback performance of a robot	CSC: RSC (Robotics service customer)
	Request maintenance of a robot	CSC: RSC (Robotics service customer)

Figure 7-2 illustrates the cloud computing sub-roles for RaaS. Figure 7-2 identifies activities of each sub-roles of RaaS which are specifically assigned to extend cloud computing services for robotics. RaaS utilizes other sub-roles of CSP and CSN to ensure the compatibility and interoperability of cloud computing.

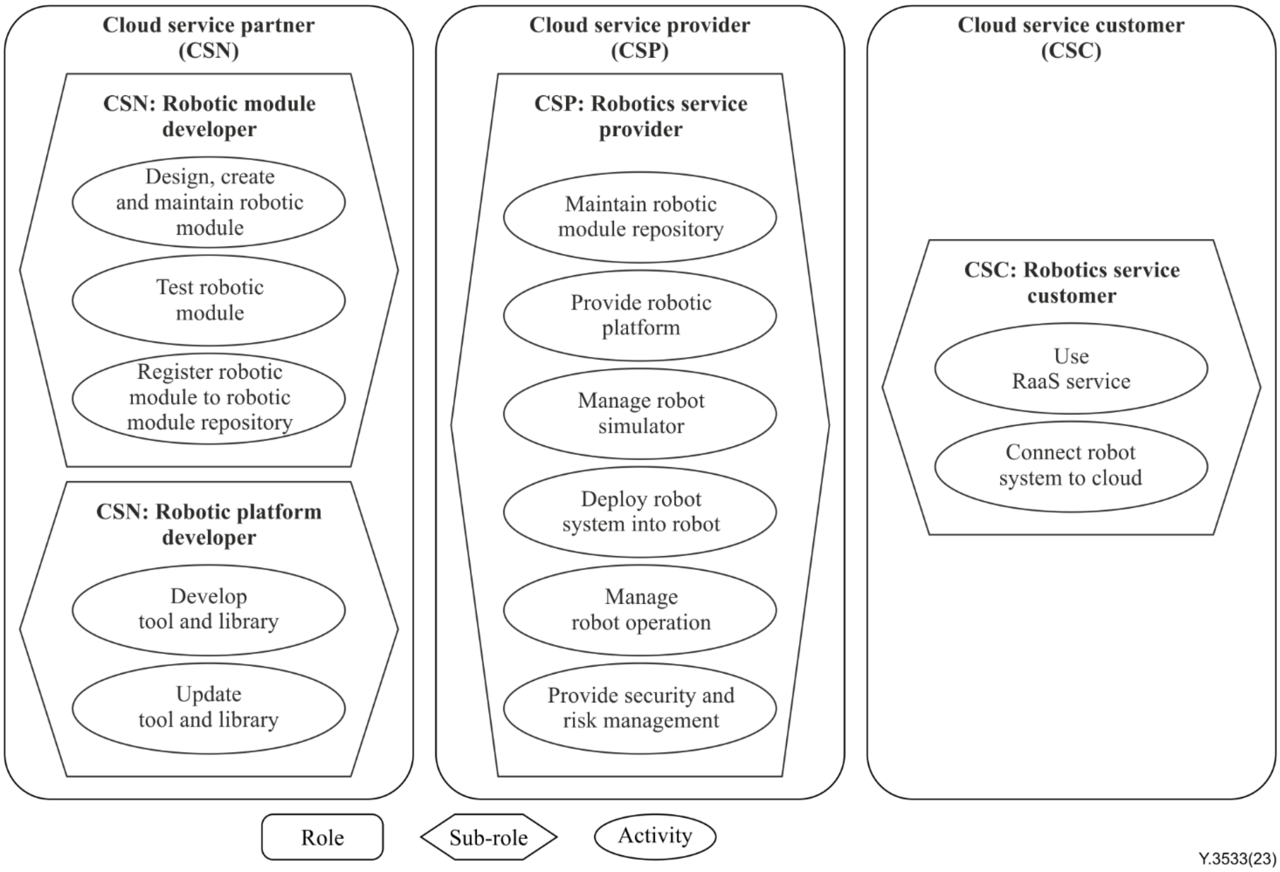


Figure 7-3 – RaaS system context

7.4 CSN:robotic module developer

CSN:RMD is a sub-role of the CSN, which provides activities related with developing robotics components. The activities of CSN:RMD involve to support to create and manage robotic module profile. The activities of CSN:RMD include:

– Design, create and maintain robotic module;

– Test robotic module;

– Register robotic module to robotic module repository.

7.4.1 Design, create and maintain robotic module

Design, create and maintain robotic module activity involves the generation of a robotic module that includes the software and hardware modules. This activity involves:

– Designing a robotic module for implementing robotic tasks or certain purpose;
– Describing a robotic module profile for a robotic module;
– Documenting the user guidelines or instructions for the robotic module;

  NOTE – The user guidelines include the guidance for composite robotic module, interface, set-up and usage instructions.

– Maintaining a robotic module with enhancements and versioning;
– Creating an interface and configurations of the composite robotic modules;
– Updating a robotic module in repository for CSP:RSP.

7.4.2 Test robotic module

Test robotic module activity involves the validation of modules with a robot simulator by cloud. This activity involves:

– Testing a robotic module with a robot simulator to ensure the execution of a robotic module correctly and completely;
– Reporting the result of simulation to CSP:RSP;
– Gathering simulation reports and valuating the simulation results to optimize a robotic module;
– Checking interoperability among robotic modules.

7.4.3 Register robotic module to robotic module repository

Register robotic module to robotic module repository activity involves describing a profile of robotic modules in the robotic module repository of cloud which is served by CSP. This activity involves:

– Describing a profile of a robotic module in the given format from CSP:RSP;
– Registering a robotic module with a profile;
– Managing a robotic module.

7.5 CSN:robotic platform developer

CSN:RPD is a sub-role of the CSN, which provides activities related to developing tools and libraries for robotics. The activities of CSN:RPD include:

– Develop tool and library;
– Update tool and library

7.5.1 Develop tool and library

Develop tool and library activity involves designing tools such as development of an integrated development environment (IDE) for robotics, and libraries such as mechanical motion calculations of robot movement. This activity involves:

– Designing a robotics development IDE;
– Designing an interface of a robot simulator;
– Developing a library for common utilities in robotics;
– Developing a library for integrating robotic modules.

7.5.2 Update tool and library

Update tool and library involves the validation of modules in the served simulation environments through the cloud. This activity involves:

– Testing compatibility of a tool and a library;
– Enhancing the functionalities of a tool and a library;
– Reflecting the issues reported on a tool and a library;
– Versioning and updating a tool and library.

7.6 CSP:robotics service provider

CSP:RSP is a sub-role of the CSP that provides RaaS capabilities including robotics and provisions of resources. The CSP:RSP focuses on making RaaS services available with interoperability among the cloud service, as well as RaaS service maintenance. The activities of CSP:RSP include:

– Maintain robotic module repository;
– Provide robotic platform;
– Manage robot simulator;
– Deploy robot system into robot;
– Manage robot operation;
– Provide security and risk management.

7.6.1 Maintain robotic module repository

Maintain robotic module repository activity involves storing robotic modules including the information of hardware (HW) modules, software (SW) modules and composite robot modules. This activity involves:

– Storing a robotic module from the CSN:RMD into the repository;
– Configurating a format of a profile of robotic module;

  NOTE – This activity involves of defining the elements of robotics profile offering with tracking the service availability, replaceability, quality, and usage.

– Providing functionalities to create, modify, and search of a robotic modules;
– Versioning and tracking history of a robotic module;
– Requesting a robotic module maintenance to the CSN:RMD.

7.6.2 Provide robotic platform

Provide robotic platform activity involves the provision of a robotic platform using cloud resources. This activity involves:

– Integrating the tools and libraries for a robotic platform;
– Provisioning resources in the cloud for a robotic platform;
– Checking compatibility of robotic modules.

7.6.3 Manage robot simulator

Manage robot simulator activity involves the provision of cloud resources for robot simulator and generation of a virtual representation for a robot. The monitoring and troubleshooting activity are provided. This activity involves:

– Performing a simulation of a robotic task with optimized cloud resources;
– Preparing a list of virtual robot representation to test the robotic systems;
– Provisioning a storage for gathering the simulation data;

– Monitoring and archiving simulation data;
– Reporting a simulation error to CSC:RSC;
– Troubleshooting a simulation error based on rules of problem resolutions;
– Providing an adjustment to support user-defined configurations for the conditions.

7.6.4 Deploy robot system into robot

Deploy robot system into robot activity involves the integration of robotic modules and development of robotic tasks for a robot system. This activity supports the packaging of the robotic modules and robotic tasks into a container image and its deployment into the robots. This activity involves:

– Testing robotic tasks in a robotic module;
– Integrating robotic modules into a robot system;
– Packaging a robot system with deployable format into container image;

  NOTE – The robot system performs a series of robotic tasks to operate a robot. The deployable format is an implementation issue which varies with the robots.

– validating a container image of a robot system in a physical robot.

7.6.5 Manage robot operation

Manage robot operation activity involves to control and manage a robot with cloud resources by implementing a robot with the adopted robotic platform. This activity involves:

– Provisioning network connectivity between cloud and a robot;
– Operating a robot with a robot system;
– Providing a user guideline to verify the risk of robot operations with cloud service coverage;
– Monitoring operations and performances of robots in the network;
– Providing a user authorization, a robot identity and a permission of control;
– Resolving a reported service fault and problem from the CSC:RSC;
– Keeping track of operation data of a robot and archiving the data.

7.6.6 Provide security and risk management

Provide security and risk management activity focuses on handling and managing risks faced by the RaaS services. This activity involves:

– Managing a rule of problem resolutions in operations and simulations of robots;
– Identifying and alerting systemic limitations in robot operations with cloud environments;

  NOTE – The systemic limitations of RaaS services with cloud includes risks and issues arisen from HW aspect of robots such as malfunctions or discrepancies of actual robot parts.

– Defining and maintaining the security and risk management policy.

7.7 CSC:robotics service customer

CSC:RSC is a sub-role of the CSC, which utilizes RaaS services to develop, simulate and operate a robot in the cloud. The activities of CSC:RSC include:

– Use RaaS service;
– Connect robot system to cloud.

7.7.1 Use RaaS service

Use RaaS service activity involves invoking and using a RaaS service.

7.7.2 Connect robot system to the cloud

Connect robot system to the cloud activity involves the operation of the robots in the real world with cloud environment. This activity involves:

– Requesting a robot system to operate with the RaaS services;

– Installing and configuring a robot system into a robot.

8 Functional requirements of cloud computing for RaaS

This clause describes the functional requirements for RaaS.

8.1 Robotic module support

The functional requirements of robotic module support include the following:

1) It is required that the CSP:RSP provide the robotic module repository.

2) It is required that the CSP:RSP provide finding robotic modules for the CSN:RMD and CSC:RSC.

3) It is required that the CSP:RSP provide templates of robotic module profiles to describe the information of robotic modules for the CSN:RMD.

   NOTE 1 – The robotic module profile provides the information of each robot such as the equipment specifications of the robot, the available robotic task, etc.

   NOTE 2 – The equipment specification provides the hardware performance or its restrictions for the robotic task. For example, the equipment specifications of robot arms describe the number of controllable arms or joints, the range of motion, the directions of axis, maximum strength, maximum speed, etc.

4) It is required that CSP:RSP provide an interface allowing the CSN:RSN to create, register, modify and manage the robotic modules within the robotic module repository.

5) It is recommended that the CSN:RSD provide the guide of robotic tasks for the robotic modules.

   NOTE 3 – The guide of robotic tasks includes the information of usage of robotic modules and the common robotic tasks that are performed by robotic modules.

6) It is recommended that the CSP:RSP provide the available robotic modules to CSC:RSC relevant with the target robotic task.

7) It is required that the CSP:RSP provide versioning for registered robotic modules for tracking and rollback of changes.

8) It is required that the CSP:RSP provide notifications to the CSN:RMD regarding malfunctions or issues related to registered robotic modules.

9) It is recommended that the CSP:RSP provide the delivery of feedback reviews from the CSC:RSC to the CSN:RMD for improvement of the robotic modules.

10) It is recommended that the CSP:RSP provide conversion of the existing cloud service applications into robotic modules for integration with cloud services provided by the CSP.

   NOTE 4 – An example of cloud services is machine learning as a service in [b-ITU-T Y.3531]. The machine learning algorithms such as object recognition are able to transform into a robotic module for enhancing the robot's ability.

11) It is required that the CSP:RSP provide security mechanism for robotic module repository to ensure the integrity and confidentiality of the robotic modules for the CSN:RMD.

8.2 Robotic platform support

The functional requirements of robotic platform support include the following:

1) It is required that the CSP:RSP provide cloud based robot simulator in robotic platform.

2) It is required that the CSP:RSP provide integrated developing environment (IDE) for the robotics to the CSN:RMD and CSC:RSC.

NOTE 1 – The IDE for the robotics applications provide robot interaction language, capabilities of building code, GUI, etc.

3) It is required that the CSP:RSP provide up-to-date tools and libraries essential for robotics in the robotic platforms.

NOTE 2 – The example of essential tools and libraries are middleware for robot operation, simulation tools, container for robot deployment, etc.

4) It is required that the CSP:RSP provide compatibility of the robotic platforms with robotic modules whenever robotic modules updates or upgrades are applied.

5) It is recommended that the CSP:RSP request maintenance of tools and libraries to the CSN:RPD.

6) It is required that the CSP:RSP provide a platform documentation, including detailed user guides, API references, and best practices, aiding both CSN and CSC in the development process.

7) It is recommended that the CSP:RSP provide visualization tools for simulating robotic tasks.

NOTE 3 – The visualization tools include the software to simulate the real-world environment with physical engines and 3D models of object for the virtual robot representation.

NOTE 4 – Simultaneous simulation of multiple robotic tasks is supported.

8) It is required that the CSP:RSP provide user-defined configurations to set simulations for a robot to CSC:RSC.

9) It is required that the CSP:RSP provide testing of the robotic modules.

NOTE 5 – The testing of robotic modules includes robotic actions, robotic tasks and robot applications.

10) It is recommended that the CSP:RSP provide import of open source tools and libraries for CSC:RSC.

11) It is recommended that the CSP:RSP provide monitoring of the robot actions.

12) It is recommended that the CSN:RMD provide the abstracted HW model for the simulation.

NOTE 6 – The abstracted HW model is representation of HW aspect and is registered in robotic module repository by CSN:RMD.

8.3 Robot deployment support

[ITU-T Y.3535] provides functional requirements for a container. The functionalities of RaaS involve container for the activities related with packaging and deployment of robotic tasks or robot systems into a physical robot. Functional requirements specified in [ITU-T Y.3535] are supported as follows:

1) It is required that a container engine execute an application in the kernel.

NOTE 1 – The container engine uses a kernel processor to run the application with the execution commands in a container file.

2) It is required that a container engine builds a container image to push to the container registry.

NOTE 2 – A container engine rebuilds the running application based on the container image and pulls it to the container registry for reuse.

3) It is required that a container engine pull container images to execute an application from the container registry.

4) It is required that a container engine provide a packaged container image for easy transport.

5) It is required that a container engine provide an image object in the container image.

NOTE 3 – An image object includes a library, file system and binary in the container file system.

The additional functional requirements of robot deployment support include the following:

6) It is required that the CSP:RSP provide the conversion of robot systems into deployable formats suitable for integration onto a physical robot.

NOTE 4 – This functional requirement differs from container services. The container images run uniformly with all dependencies across robot hardware not considering the change of a hardware part on a robot.

7) It is required that the CSP:RSP provide field test suites for enabling the CSC:RSC to validate robotic tasks under real conditions.

8) It is required that the CSP:RSP provide reporting malfunctions or unexpected issues promptly to the CSC:RSC.

9) It is recommended that the CSP:RSP provide a version control to manage different versions of deployed robotic modules and robot systems.

10) It is recommended that the CSP:RSP provide rollback to allow CSC:RSC to return to previous versions of deployed robotic modules.

11) It is recommended that the CSP:RSP provide monitoring tools for real-time tracking by the CSC:RSC.

8.4 Robot operation support

The functional requirements of robot operation support include the following:

1) It is required that the CSP:RSP provide network connections upon request between CSP:RSP and CSC:RSC.

2) It is required that the CSP:RSP provide stable network protocols for operating robots.

3) It is recommended that the CSP:RSP provide the encapsulation of robot control massage for security.

NOTE 1 – The encapsulation encodes the robot control message with protocols that encrypt and authenticate the message.

4) It is required that the CSP:RSP provide real-time monitoring tools to track the status, battery lifetime and performance metrics of operational robots for CSC:RSC.

5) It is required that the CSP:RSP provide tools for debugging or troubleshooting, assisting CSC:RSC in identifying operational issues in real time.

6) It is recommended that the CSP:RSP provide lifecycle management of robotic tasks.

NOTE 2 – The lifecycle management can be implemented with the graphical user interface for monitoring the robotic tasks.

7) It is required that the CSP:RSP provide scalability for the simultaneous operation of multiple robots.

8) It is required that the CSP:RSP provide a storage for log data of robot operations, and reports including operational anomalies or failures.

9) It is recommended that the CSP:RSP provide a maintenance support for operated robots.

NOTE 3 – The maintenance support includes updating robot software and firmware remotely, and notifying worn-out parts if detected.

10) It is required that the CSP:RSP provide risk management to use cloud service for robot operation.

NOTE 4 –Risk management includes an emergency stop and fail-safe mechanism to recover the robot operations.

9 Security considerations

It is recommended that the security framework for cloud computing described in [b-ITU-T X.1601] be considered for the robotics as a service. [b-ITU-T X.1601] analyses security threats and challenges in the cloud computing environment and describes security capabilities that could mitigate these threats and meet security challenges.

[b-ITU-T X.1631] provides guidelines supporting the implementation of information security controls for cloud service customers and cloud service providers. Many of the guidelines guide the cloud service providers to assist the cloud service customers in implementing the controls and guide the cloud service customers to implement such controls. Selection of appropriate information security controls and the application of the implementation guidance provided, will depend on a risk assessment as well as any legal, contractual, regulatory or other cloud-sector specific information security requirements.

Relevant security requirements of [b-ITU-T Y.2201], [b-ITU-T Y.2701] and applicable X, Y and M series of ITU-T Recommendations need to be taken into consideration, including access control, authentication, data confidentiality, data retention policy, network security, data integrity, availability and privacy.

Security considerations for RaaS in this Recommendation pertain exclusively to cloud computing aspects. Security issues for robot-related aspects including the physical safety of robot, risk management and operational integrity of robot systems are referenced through [b-ISO 22166-1], [b-ISO 10218-2], [b-ISO 12100], and [b-ISO 13482]. It is recommended that information of robot-related security considerations be provided to ensure RaaS stakeholders are aware of the coverage of security in cloud computing.

Appendix I

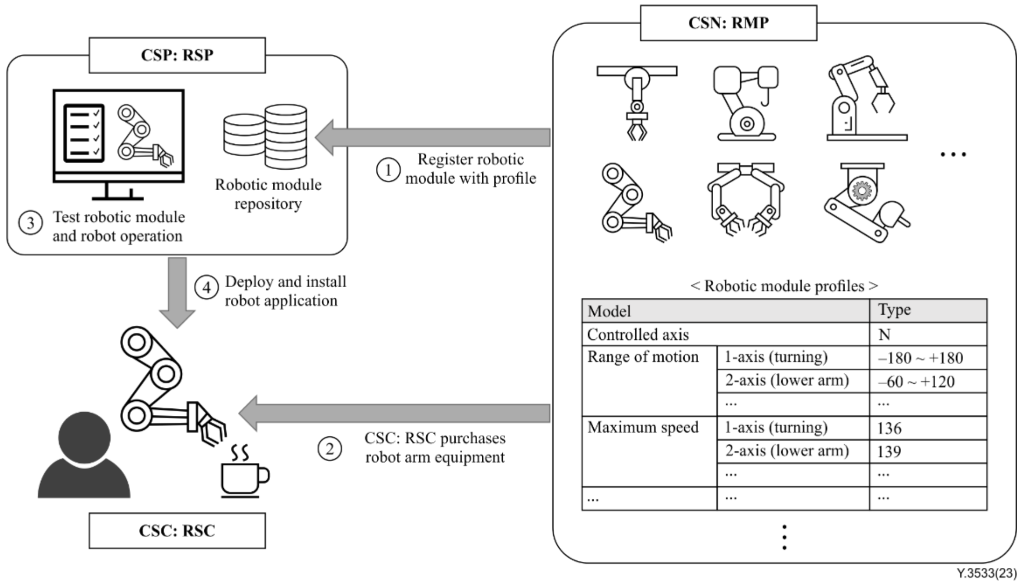
Use case of robotics as a service

(This appendix does not form an integral part of this Recommendation.)

The use cases in this appendix provide examples of RaaS with the functionalities of robotics related functional requirements of RaaS.

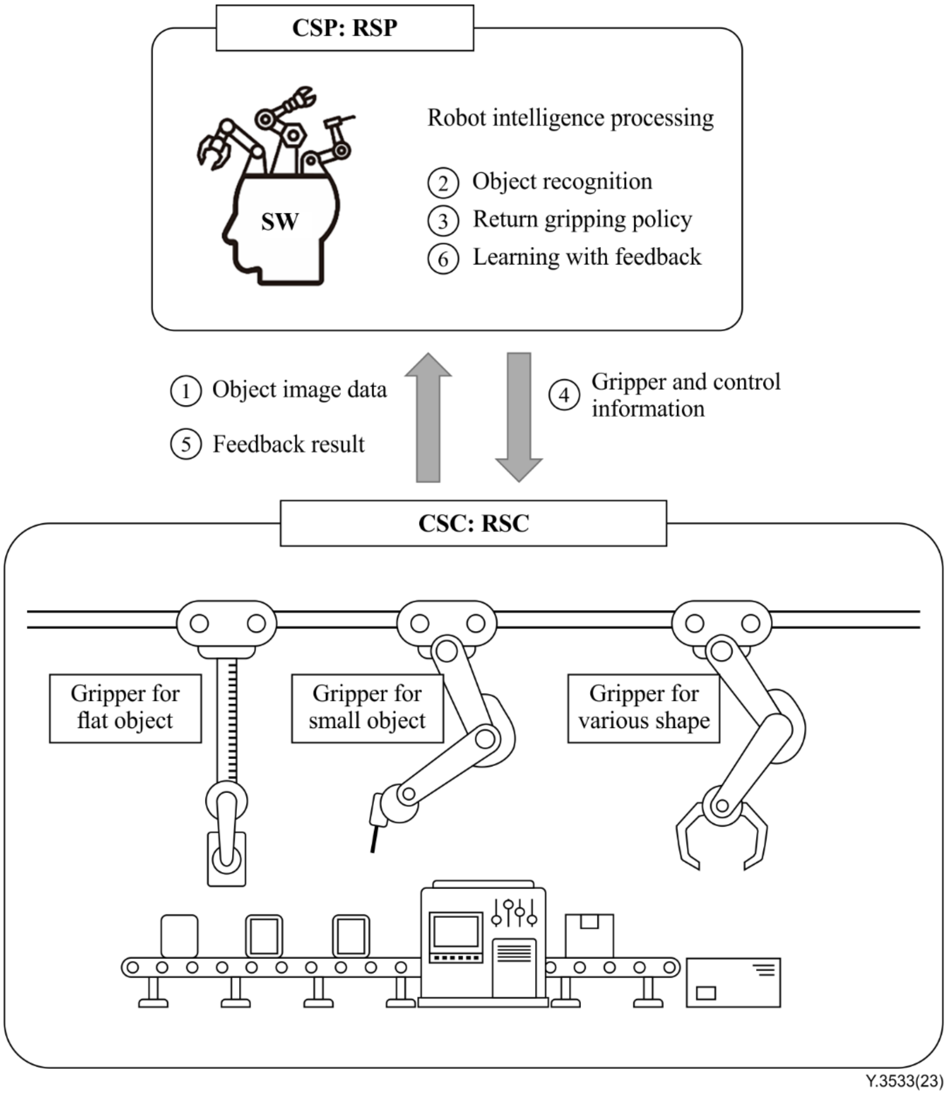
I.1 Registration of robotic module profiles and development of robot applications using the profiles

Title	Registration of robotic module profiles and development of robot applications using the profiles.
Description	This use case describes procedures of developing robot applications with registration of robotic module profiles. For developing robot applications, the robotic task should be designed with considerations of robot equipment in its profiles and their constrictions. For example, if the robot developer wants to develop robot baristas that can make coffee with robot arms, they need to consider the specification of robot arms such as range of motion, speed of motion, grip strength, etc. The following are general steps for developing the robot applications from registering robotic modules to using robot applications in cloud: 1) CSN:RMD registers robotic modules in the robotic module repository which is managed by CSP:RSP. 2) CSP:RSP runs robotic platform and prepare robot profile repository. 3) CSC:RSC develops the robot applications with robotic modules in robotic platform environment. A. CSC:RSC requests the robotic modules for certain robotic tasks. NOTE – CSP:RSP prepare the list of robot tasks for certain robotic tasks by gathering feedback or reports of its usages from both CSN:RMD and CSC:RSC. B. CSC:RSC designs the robot applications with using robotic platforms. C. CSC:RSC requests to test the performance of robot applications and package the developed applications. 4) CSC:RSC install robotics application.
Role/Sub-role	CSN:RMD (CSN:Robotic Model Developer) CSP:RSP (CSP:Robotics Service Provider) CSC:RSC (CSC:Robotics Service Customer)

Figure (optional)	
Pre-conditions (optional)	CSC:RSC purchases robot equipment from CSN:RMD. CSN:RPD provide the robotic platform supporting compatibility of robotic modules. CSP:RSP provide the computing resources for robotic platform
Post-conditions (optional)	CSC:RSC feedback the result of robot application deployment to CSP:RSP.
Derived requirements	– Clause 8.1 items 1), 2), 3), 4), 5), 6) – Clause 8.2 items 1), 2), 3), 4) 6) – Clause 8.3 items 2), 3), 4), 5)

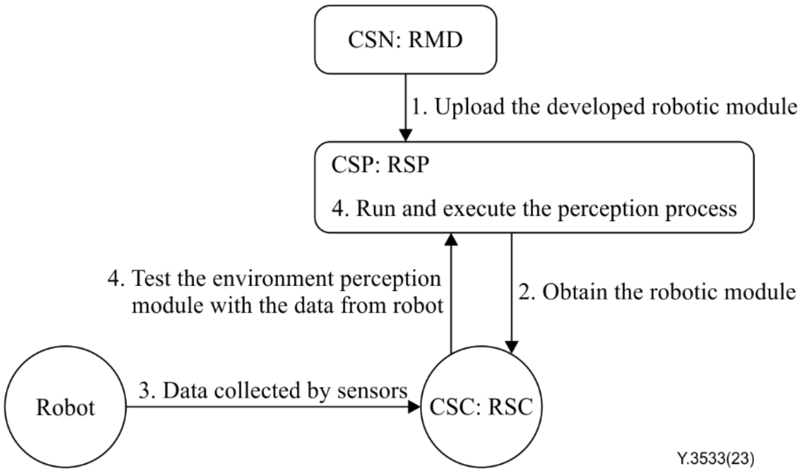
I.2 Operation of robot intelligence using RaaS in the environment of an industrial roller conveyor with multi-gripper

Title	Operation of robot intelligence using RaaS in the environment of an industrial roller conveyor with multi-gripper.
Description	This use case provides the industrial robot system with multiple grippers in a RaaS environment. The multiple grippers are used for supporting various objects in the real world. The robot intelligence may be required high amounts of resources to utilize the various types of grippers. It supports to control the multiple grippers in given conditions, for example, the two-finger gripper adapted to grip stick-shaped object such as screwdriver, but not to grip a flat object. Multiple grippers can support the various objects in the manufacturing industry. The gripping tasks are executed by the robot intelligence which is implemented by a machine learning algorithm in the cloud. The following are the steps of robot intelligence processing for operating grippers with the real field data: 1) CSC:RSC connects multiple grippers of the robot to the robot intelligence in the cloud. 2) CSC:RSC transmits the object image data to CSP:RSP. 3) CSP:RSP executes and predicts the object and return the result of gripping plan. 4) CSP:RSP sends the gripper type and control message to CSC:RSC. 5) CSC:RSC executes the gripping tasks and return feedback to CSP:RSP. 6) CSP:RSP updates the gripping status of the robot with feedback.
Role/Sub-role	CSP:RSP (Robotics service provider) CSC:RSC (Robotics service customer)

Figure (optional)	
Pre-conditions (optional)	CSN:RMD designs and publishes robotic model for the robot intelligence algorithm to using multiple gripper. CSP:RSP runs the robotic platform for robot operations in advance.
Post-conditions (optional)	CSP:RSP computes the robot intelligence algorithms continuously until user's request to stop.
Derived requirements	– Clause 8.2 items 3), 4), 6), 7), 11) – Clause 8.3 items 1), 2), 3), 6), 7), 9), 10), 11) – Clause 8.4 items 1), 2), 3), 4), 5), 6), 7), 8), 10)

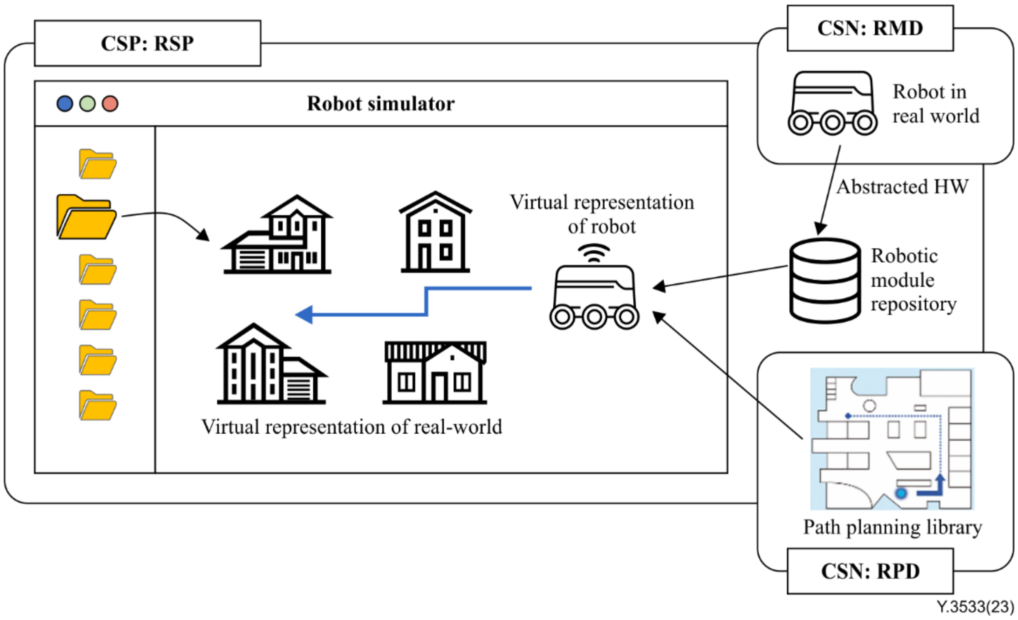
I.3 Enhancement of environment perception of a robot using RaaS

Title	Enhancement of environment perception of a robot using RaaS.
Description	Environment perception aims to endow a robot capability of seeing and understanding its environment, such as being aware of the nearby objects with locations and size. It is a fundamental step for further robotic tasks in robot system. The key technologies of environment perception usually involve image segmentation, object detection, etc., which is suitable to be implemented in the cloud. The interactions of using RaaS service for this use case include the following steps: 1) CSN:RMD develops perception SW of robotic module and register to the CSP:RSP. CSP:RSP validates and stores the robotic module. 2) In the process of developing a robot with environment perception capability, CSC:RSC obtain the environment perception module from CSP:RSP.

	3) CSC:RSC acquires data from robot sensors such as 2D camera, 3D camera or 3D Lidars, and runs the environment perception modules to analyse the data. Calibration method for data might also be needed according to the image data provided by CSP:RSP. 4) CSC:RSC performs testing the environment perception of robot in test suite designed by CSN:RPD. 5) After completing the test, CSC:RSC integrates the environment perception module into robot system by using packaging library from CSP: RSP.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Clause 8.1 items 1), 2), 3), 4), 5), 7), 8), 9), 10), 11) – Clause 8.2 items 1), 2), 4), 5), 7), 8), 9), 11), 12) – Clause 8.3 items 1), 2), 3), 6), 7), 9), 10), 11)

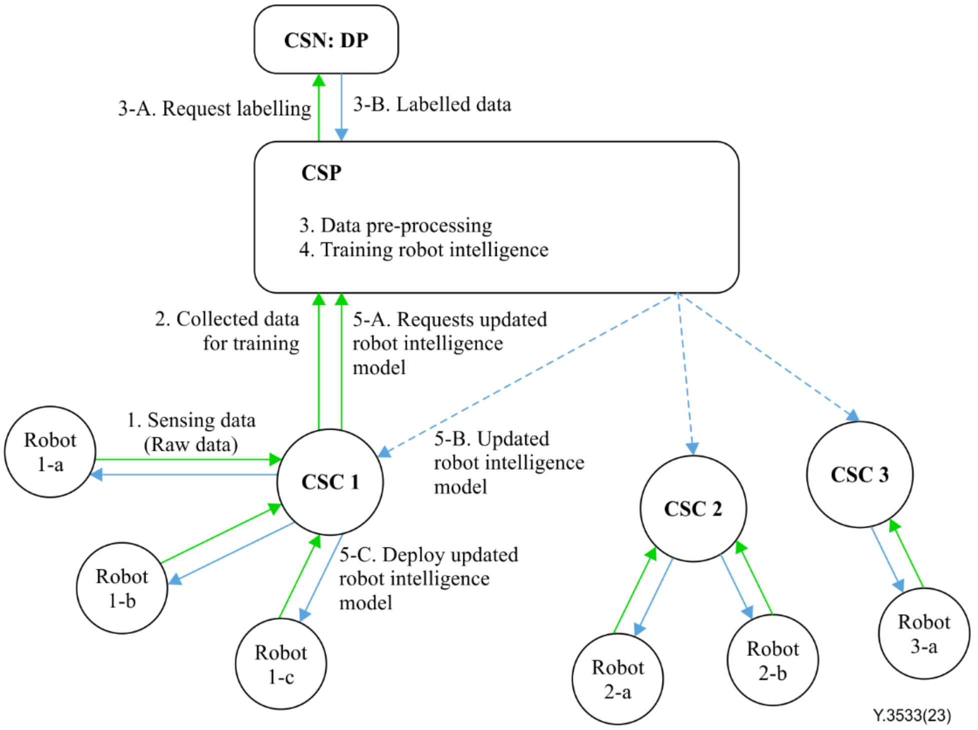
I.4 Developing and testing a robot system with a fine-tuning navigation module

Title	Developing and testing a robot system with a fine-tuning navigation module.
Description	This use case provides the developing and testing related functions of RaaS. The CSC:RSC wants to develop the delivery robots which requires navigation systems for achieving the robotic task. CSC:RSC search the navigation module and fine-tune it in the RaaS environment. The CSC:RSC builds the robotic system which executes the go-to-delivery task by composing some HW and SW modules from the robotic module repository. The robot simulator simply executes the robot system only for testing the navigation task with virtual robot representation. The following are the summarized steps of developing and testing robot system: 1) CSC:RSC searches the HW and SW modules for composing a delivery robot system in robotic module repository. 2) CSC:RSC requests related robotic modules including path planning module, and mobility control module for developing navigation module. 3) CSP:RSP provides the robot simulator for developing and testing to CSC:RSC 4) CSC:RSC configures gathered modules and tunes the composed modules to derive expected navigation task.

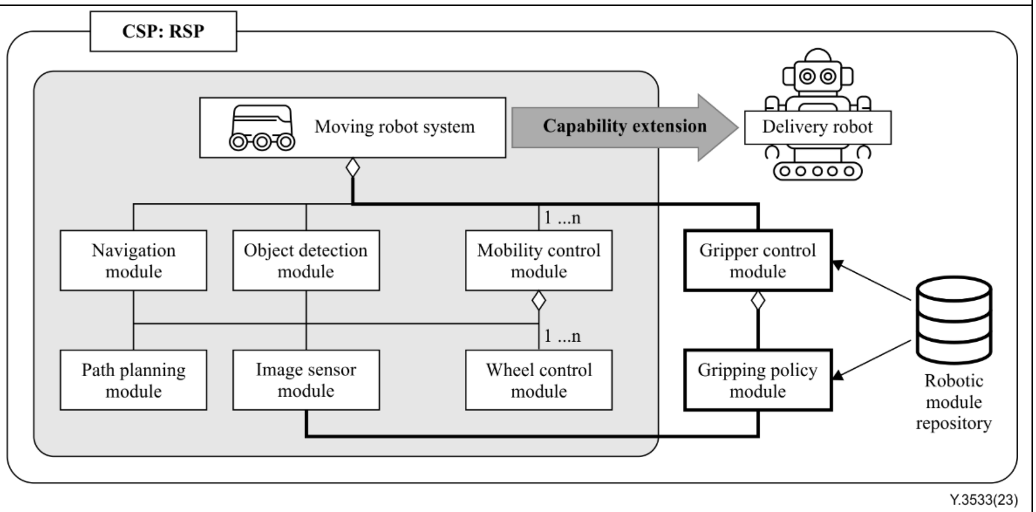
	5) CSP:RSP provides the virtual representations of robot which is abstracted HW parts with physical engines for virtual world which is designed by CSN:RPD. 6) CSC:RSC tests their robot system with monitoring the implementations of navigation, and get the log of test results from CSP:RSP.
Role/Sub-role	CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Clause 8.1 items 1), 2), 5), 6), 7), 10) – Clause 8.2 items 1), 2), 4), 6), 7), 8), 9), 10), 11), 12)

I.5 Deployment of the distributed robots using RaaS & cloud service

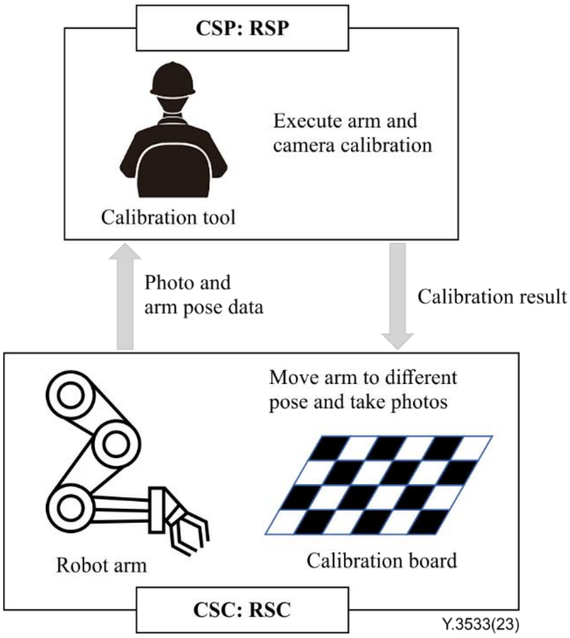
Title	Deployment of the distributed robots using RaaS and cloud service.
Description	In this use case, the multiple robots are distributed across multiple locations. The robot intelligence for the robots is trained by collected data in different locations. The CSC:RSC can update their robot intelligence without stopping their robot operations by using RaaS. The following are specific procedures for updating and deploying robot intelligence: 1) CSC:RSC collects the raw data from their robots including sensor data, motor data, and related data. 2) CSC:RSC send the raw data to CSP:RSP for training their robot intelligence which is previously deployed in the cloud. 3) CSP:RSP performs the data pre-processing including data cleaning, and labelling with machine learning application in cloud service. A. CSP:RSP requests labelling task to CSN:data provider which is the sub-role of machine learning service in cloud. B. CSN:DP provides labelled data to CSP:RSP. 4) CSP:RSP trains robot intelligence with the labelled data from CSN:data provider. 5) CSP:RSP updates robot intelligence and versioning the model.

	6) CSP:RSP deploys robot intelligence when the request is occurred from CSC:RSC. A. CSC:RSC requests the updated models to CSP:RSP. B. CSP:RSP deploys the updated models to CSC:RSC with over-the-air mechanism. C. CSC:RSC operates the updated model in their robots.
Role/Sub-role	CSN:data provider CSP:RSP (CSP:Robotics Service Provider) CSC:RSC (CSC:Robotics Service Customer)
Figure (optional)	
Pre-conditions (optional)	CSP:RSP provides the functionalities of machine learning as a service for the RaaS service.
Post-conditions (optional)	
Derived requirements	– Clause 8.2 items 4), 8), 9) – Clause 8.3 items 1), 2), 3), 4), 5), 6), 7), 9), 11) – Clause 8.4 items 1), 2), 3), 4), 6)

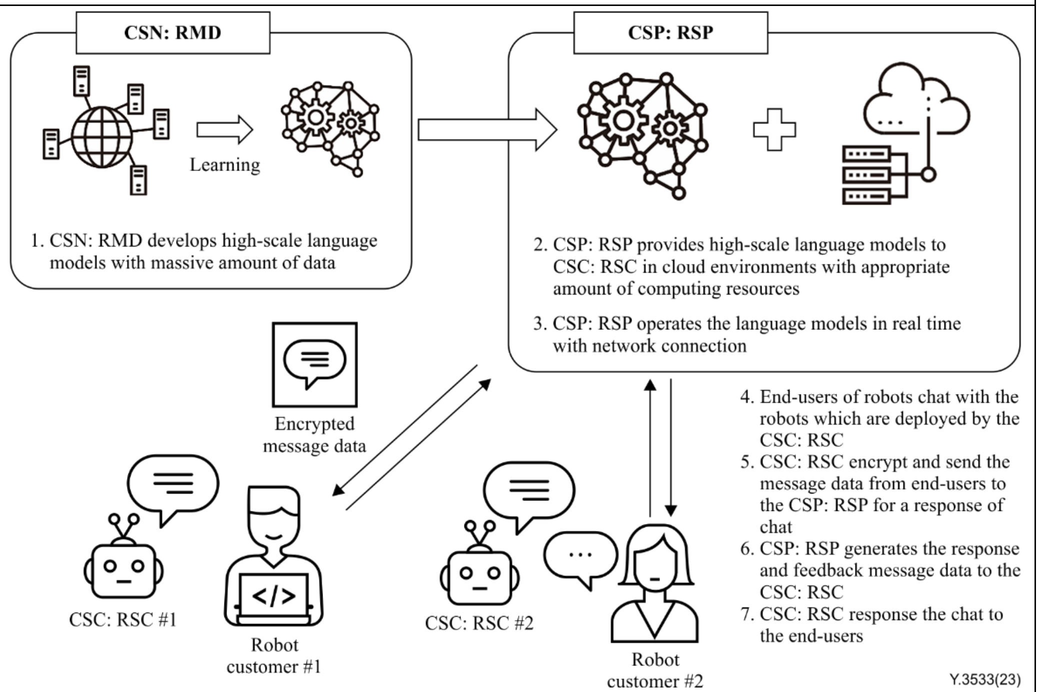
I.6 Extension of robot system with adding robotic modules

Title	Extension of a robot system by adding robotic modules
Description	CSC:RSC wants to extend their delivery robot to have additional abilities to grip the objects and load the objects. Rather than developing the new robotic module for gripping, the CSC:RSC searches the pre-designed robotic modules for new abilities in the robotic module repository. The CSC:RSC finds the gripper control module and gripping policy module. The CSC:RSC then integrates them into their delivery robot system manager as they are already deployed in the robot. The input/output data and proper interfaces for new modules are provided by CSP:RSP for executing the newly updated gripping task. After validating the extended robotic task in the robot, CSC:RSC reports the reviews of newly updated capabilities of gripper control module to CSP:RSP. The CSP:RSP feed them back to the CSN:RMD who have the authority of that module.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	The CSC:RSC builds the robot system in advance with the proper format which is compatible with the format in the RaaS environment. The CSN:RPD provides the proper interface and input/output data to control the robotic modules.
Post-conditions (optional)	The CSP:RSP provides the testing suites for the extended robotic task for CSC:RSC.
Derived requirements	– Clause 8.1 items 1), 2), 4), 6), 8), 9), 10) – Clause 8.2 items 1), 2), 4), 5), 6), 7), 8), 9), 10), 12) – Clause 8.3 items 9), 10)

I.7 Fine-tuning of robotic modules with recalibration in RaaS service

Title	Fine-tuning of robotic modules with recalibration in RaaS service
Description	This use case describes how CSC:RSC use the calibration software provided by CSP: RSP to fine-tune their robots. In this use case, the robot of CSC:RSC calculates the relative position and direction of arms with cameras, so that the robot can accurately locate objects and plan their operation routes through the equipped cameras. In this use case, the robot of CSC:RSC needs to calculate accurate position and direction of arms with the conditions gathered by cameras. However, the deployed robot is not accurate enough to grip the object correctly. The CSC:RSC recalibrates the robot by using the calibration library provided by CSP:RSP in the cloud. This use case includes the following steps: 1) CSC: RSC prepares a calibration board and places it in an appropriate position; 2) CSC: RSC sends a message to the cloud indicating the start of the calibration process; 3) CSC: RSC moves the arm and takes photos of the calibration board in different positions and directions; 4) CSC: RSC sends photos of the arm and corresponding posture (position and direction) to the cloud; 5) CSP: RSP runs calibration library using data provided by CSC:RSC; 6) CSP: RSP returns the calibration results to CSC:RSC; 7) CSP: RSP stores the results that will be used for further operations.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	The robot of CSC:RSC is equipped with cameras which have sensors of depth and RGB stream.
Post-conditions (optional)	
Derived requirements	– Clause 8.1 items 1), 2), 5), 7), 8), 11) – Clause 8.2 items 1), 2), 4), 5), 6), 8), 11), 12)

I.8 Remote control and offloading robots in a cloud environment via network

Title	Remote control and offloading robots in a cloud environment via network
Description	In this use case, the CSC:RSC controls robots remotely and offload robot intelligence in a cloud environment when a network connection is supported between robots and cloud. The CSP:RSP supports access to the robots from anywhere with an Internet connection. The CSC:RSC utilizes cloud resources to implement advanced intelligence models, such as large-scale language models. Due to resource constraints within the robot, the CSC:RSC offloads tasks to the cloud to leverage its computing capabilities. The figure shows an example of the remote control and offloading robot with chat systems. The following flow is an execution of the robot chat systems with RaaS service. 1) CSN:RMD develops high-scale language models with massive amount of data and register it in robotic module repository. 2) CSP:RSP provides high-scale language models to CSC:RSC with appropriate amount of computing resources. 3) CSP:RSP operates the language models in real-time with network connection. 4) The robots chat with the robots which are deployed by the CSC:RSC. 5) CSC:RSC encrypt the message data of robots and send it to the CSP:RSP for a response of chat. 6) CSP:RSP generates the response and feedback message data to the CSC:RSC. 7) CSC:RSC response the chat to the user of robot. The security and privacy issues using cloud-based controlling is described in the execution. The privacy sensitive data may be transmitted over the internet. CSP:RSP should ensure the reliable and secure RaaS service by supporting encryption of data. CSP:RSP should prevent any disruption which affect the operation of the robots.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	The CSP:RSP provide IDE to develop high-scale language models for CSN:RSD.

Post-conditions (optional)	The CSP:RSP provides the flexible computing resource capabilities for adopting the upgrade of the language models. The CSN:RMD supports to manage and update the language models to upgrade its performance of quality.
Derived requirements	– Clause 8.1 items 1), 2), 4), 7), 9), 10), 11) – Clause 8.2 items 3), 4), 5), 11) – Clause 8.3 items 1), 2), 3), 4), 5), 6), 11) – Clause 8.4 items 1), 2), 3), 4), 5), 8), 9), 10)

Appendix II

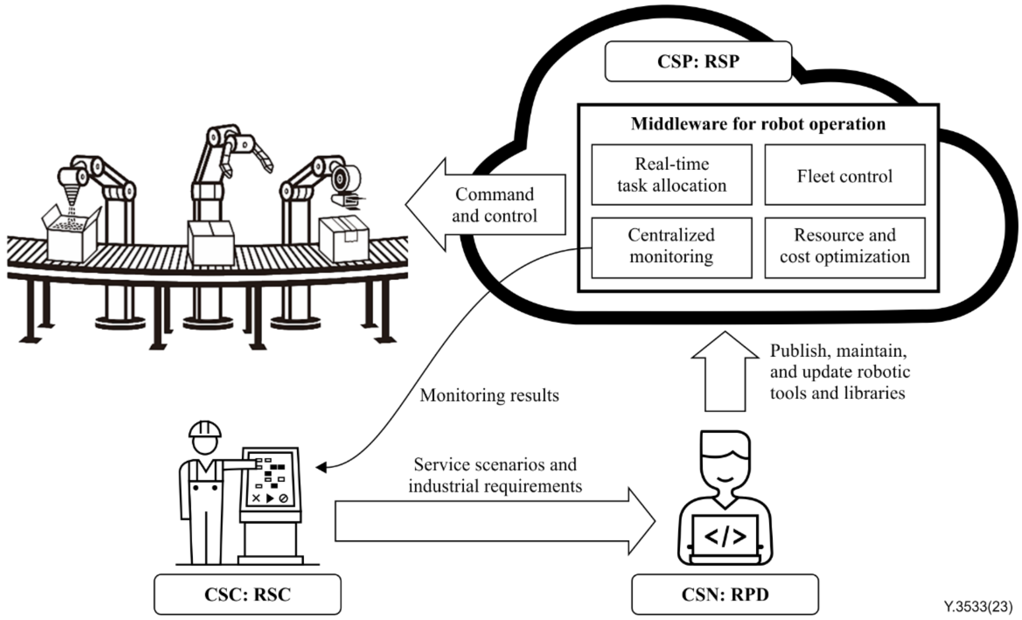
Use case of robotics as a service for application perspective

(This appendix does not form an integral part of this Recommendation.)

Appendix II includes a few examples of RaaS in different applications or industrial domains. The specific functional requirements in each domain will vary depending on the type of application, deployment of the robots, or robot regulations in each industrial domain. In the RaaS, the mandatory and optional requirements vary based on the conditions of the operation environment.

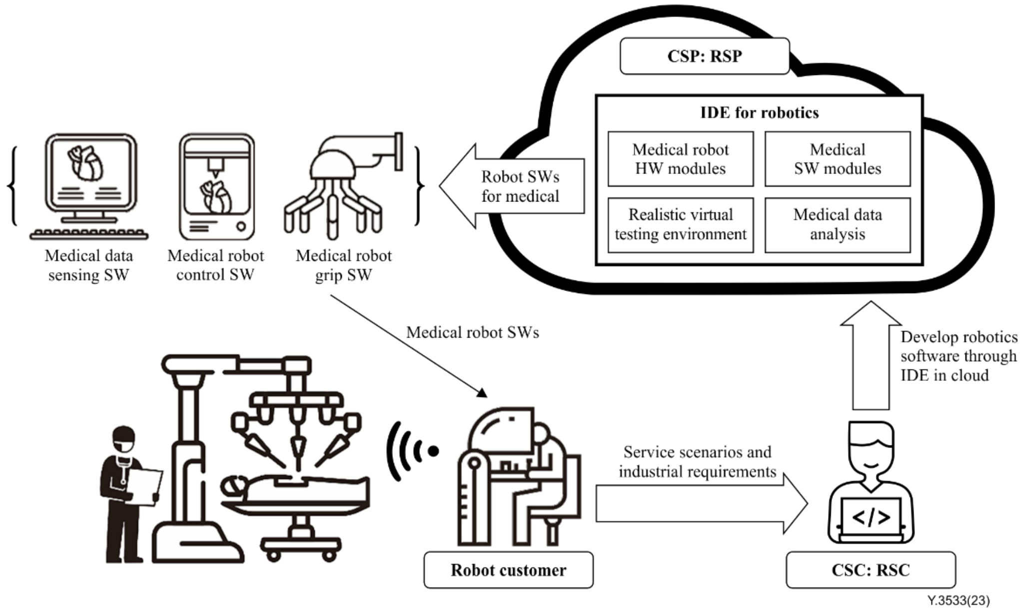
II.1 Operations of robots with cloud and network connectivity in warehouse environment

Title	Operations of robots with cloud and network connectivity in warehouse environment
Description	This use case describes operations of robots in warehouse environment. The operations of robots are supported by robotic platform and RaaS services with network connection, which include the computational processing of robot's movement through the cloud. In this use case, robots of CSC:RSC are connected with the network and the CSP:RSP supports the low latency services for normal robot operation. The overall computation of the execution for the tasks are processed by the computational resources in the cloud. In this scenario, the robots allocated in the warehouse are only equipped with sensors, actuators and minimal hardware enabling them to receive commands from the cloud and transmit sensor data to the cloud. The figure shows the overall flows of this scenario. The CSC:RSC have some industrial requirements for robot operations and the CSN:RPDs develop and publish appropriate tools and libraries for the requirements. In this use case, the following tools and libraries for robotics are considerable as examples: • Real-time task allocation library: The data collected by the robots are sent to the cloud for processing. The robot performs algorithms to analyse the data and provide real-time commands or control messages for each deployed robot. • Centralized monitoring tool: The centralized monitoring is provided to control the robots. Real-time analytics are used to monitor the performance of the robots, identify opportunities for improvement, and make adjustments as needed. • Fleet control library: The centralized control and monitoring enables the effective fleet control. This includes tracking the location and status of each robot, as well as assigning tasks and optimizing routes based on real-time environment. • Resource and cost optimization library: By analysing data on demand, the library is used for optimizing resource allocation to minimize waste of operational costs.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)

Figure (optional)	
Pre-conditions (optional)	The CSP:RSP provides a network connection to the CSC:RSC and supports enough network latency for operating robots in real-time.
Post-conditions (optional)	The performance of the robots in the warehouse is monitored, and the tasks are performed more efficiently or the operational costs are optimized.
Derived requirements	– Clause 8.2 items 3), 4), 5), 10), 11) – Clause 8.3 items 1), 2), 3), 4), 5), 7), 8), 9), 10), 11) – Clause 8.4 items 1), 2), 3), 4), 5), 6), 7), 8), 9), 10)

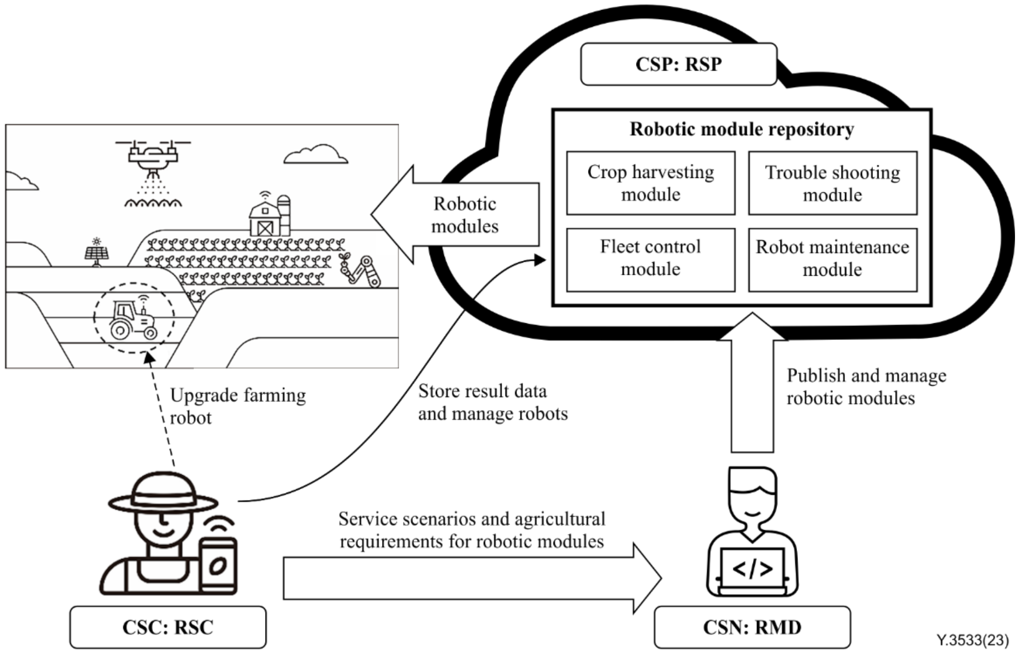
II.2 Medical robot software development on the cloud environment

Title	Medical robot software development on the cloud environment
Description	This use case provides the development scenario for medical robot software in the cloud environment. In this use case, the user of the robot is a doctor who wants to deploy and perform remote surgery with a robot. In this use case, CSP:RSP supports the following functionalities or services through the cloud: • Medical robot HW modules: The IDE on the cloud supports tools for utilizing the HW modules for medical robots. This can include 3D-designed simulation tools, and specialized command or control of medical sensors. • Medical SW modules: The IDE also includes tools and resources for developing SW modules for medical robots. This can include programming languages and libraries, as well as specialized software for medical imaging and analysis. • Realistic virtual testing environment: To simulate the operation of medical robot SW before final publishing, the CSC:RSC test their SW in a realistic virtual testing environment. This can include virtual models of medical environments, such as surgical rooms or patient wards, as well as simulation software to check the safety issues. • Medical data analysis: The cloud service for collecting and analysing medical data can be provided. This can include machine learning algorithms specialized in medical data analysis. Real-time support is possible if the user requests the service. By leveraging these IDE functionalities developers can easily design, develop and test medical robot SW such as medical data sensing SW, medical robot control SW and medical robot grip SW. The user of medical robots can utilize these SW modules for operating medical robots.

Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform Developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Clause 8.1 items 1), 3), 4), 5), 7), 8), 9), 11) – Clause 8.2 items 1), 2), 3), 4), 5), 6) – Clause 8.3 items 1), 2), 3), 4), 5), 7), 8), 9)

II.3 Operating agriculture robot system with cloud management

Title	Operating agriculture robot system with cloud management
Description	This use case describes operations of a robot system in an agricultural environment with cloud management. In this use case, the CSC:RSC who operates farming robots considers potential network failures caused by weather or obstacles, so the CSC:RSC deploys the robot system in a way that allows the robot to operate independently without network connections. The robots of CSC:RSC can be freely located in farms or forests to harvest or manage crops. The robots automatically transmit collected data to the cloud when a network connection is available, and the cloud also deploys updated robot systems to the robots via a maintenance service. The figure illustrates this scenario. The CSN:RMD develops and publishes robotic modules for agriculture and harvesting robots. The CSC:RSCs the utilize these robotic modules in their robots and manage them using the cloud environment. CSP:RSP supports robot management. The following are the robotic modules provided by CSP:RSP: • Crop harvesting module: Crop harvesting module is operated with sensors and cameras navigate through the fields to collect data on crop growth and health. The data are sent to the cloud for analysing the information of crop status for improving efficiency or making decision for harvest. • Trouble shooting module: The trouble shooting module monitors the performance of the robots and detect errors or malfunctions, such as battery life or sensor failures. When an error or problem is detected, the management system can send alerts and notifications to farmers to troubleshoot and resolve issues.

	• Fleet control module: The fleet control provides centralized control and monitoring of the robot fleet, including tracking the location and status of each robot and optimizing routes based on collected data. This enables operators to allocate and assigns tasks to the robots more efficiently. • Robot maintenance module: The maintenance provides software updates to the robots, ensuring the robot can adopt the latest technology and algorithms.
Role/Sub-role	CSN:RMD (CSN:Robotic model developer) CSN:RPD (CSN:Robotic platform developer) CSP:RSP (CSP:Robotics service provider) CSC:RSC (CSC:Robotics service customer)
Figure (optional)	
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirements	– Clause 8.1 items 1), 3), 4), 5), 10), 11) – Clause 8.2 items 1), 2), 3), 4), 5), 6), 7), 8), 9), 11) – Clause 8.3 items 1), 2), 3), 4), 5), 7), 8), 9), 10), 11) – Clause 8.4 items 1), 2), 3), 4), 5), 6), 7), 8), 9), 10)

Appendix III

Technical Concepts of modularity, virtualized simulation environment and robotic system deployment in robotics

(This appendix does not form an integral part of this Recommendation.)

Appendix III is provided to support the conceptual understanding of robotics and related terms associated to robotic modules, systems and platforms.

III.1 Robotic modules and modularity in robotics

Robot development is an integrated technology that combines various complex technologies including hardware design, system control, machine learning and others. The development of each technology may need expertise in various fields belonging to separate research domains. Modular-based development has many advantages such as:

– Faster time for developing;

– Reduced costs with re-usability;

– Safety benefits with standardizations;

– Higher collaboration;

– Improved manageability.

NOTE 1 – Cloud-computing can support the advantages of modular-based robotics with composed services and resources.

The term "module" in a general sense refers to a part of a system, particularly in the context of software or hardware. It is a unit that performs a specific function and can operate independently, but which is designed to work as a part of a larger system. Modules make systems easier to manage and understand, as each module can be developed, replaced or updated independently. In ITU, a "module" is not defined on its own but in many domains including software module, security module, battery module, and others.

A "robotic module" specifically refers to a module that is a part of a robotic system. This could be a hardware, such as an actuator, sensor or manipulator, that contributes to the robot's physical capabilities, or software, such as a control algorithm or data processing routine, that contributes to the robot's abilities.

In robotics, robotic modules provide the configurations of the hardware and software aspects with the information including the types of modules, their manufacturer and the connectors or interfaces.

NOTE 2 – In [b-ISO 22166-1] and [b-ISO 22166-201], module and related terms are defined without 'robotic'. The terms are defined as:

- **Module** [b-ISO 22166-201]: component or assembly of components with defined interfaces accompanied with property profiles to facilitate system design, integration, interoperability, and re-use
- **SW module** [b-ISO 22166-201]: module whose implementation consists purely of programmed algorithms
- **HW module** [b-ISO 22166-201]: module whose implementation consists purely of physical parts, including mechanical parts, electronic circuits and any software, such as firmware, not externally accessible through the communication interface
- **HW-SW module** [b-ISO 22166-201]: Module whose implementation consists of physical parts, including mechanical parts and electronic circuits, and any software that is accessible through the communication interface

The modular framework shall design and define the robotics service in the virtualized environment for simulating the robot behaviours. The module-based framework may support the reusability, interoperability and composability of the modules with well-defined operation mechanism, communication protocols, interfaces with supporting data types. Those aspects are shared and exchanged with the profiles including the values of module properties. The module properties shall be defined for proper execution.

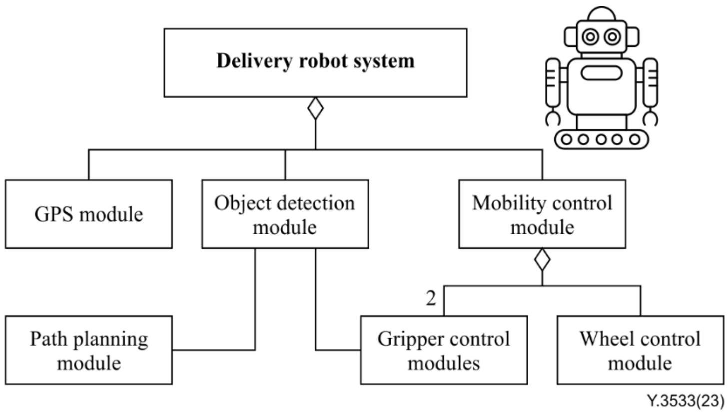


Figure III.1 – A conceptual configuration for module-based robotics service of delivery robot

Figure III.1 shows a conceptual configuration of modules with hardware and software aspects for implementing a delivery service robot. Its main behaviours may be the following:

- Recognize the commands from authorized individuals;
- Move to the specific desired location;
- Identify objects and avoid potential hazards during travel.

To execute both behaviours, the SW modules and the HW modules for a delivery service robot are classified into an identification module, data exchange modules, a security module, a navigation module, an obstacle avoidance module, a safety module, and mobility control modules.

The identification module consists of a personal identification module for face recognition of authorized individuals and an object identification module for recognizing safety-related objects. The data exchange module is used to exchange data among the delivery robot, servers and other robots as appropriate. The commands are encrypted and can be delivered and read by modules having the correct authority provided by a security module. The navigation module consists of a mapping module, a localization module and a path scanning module to control the mobility and check the path and obstacles. The safety module manages the hazards faced by the robot, including security risk considerations. The mobility control modules are HW modules for controlling the four actuators in the wheel modules.

III.2 Virtual robot representation

The virtual robot representation is a replicated world used for developing robot applications due to the limitation of robot hardware availability. For training robot behaviour, the iterative tests and validations are performed. If the engineer tests their robots in the real world, damage to the robot may occur during the testing process, and too much time is consumed in performing even a single-task. Therefore, testing a robot in the real world without simulation requires enormous resources in cost and time.

To replicate the real world in a virtual representation, the robot simulator supports the 3D-models of objects or robots, physics engine with mathematical representations, and a 3D-rendering engine with realistic movements with graphics. The following are the explanation and example of each component for simulator:

- 3D-Models: Mathematical representations of an object's physical body in 3D-space;

  NOTE 1 – Examples of 3D-models formats are standard digital format (SDF), unified robot description format (URDF), etc.

- Physics engine: Realistic movements with mathematical representations;

  NOTE 2 – Examples of physics engine are ODE, Bullet Simbody, etc.

- 3D rendering engine: Graphics for the environments.

  NOTE 3 – An example of engine for 3D rendering is the object-oriented graphics rendering engine (OGRE).

III.3 Robot system deployment: robot-server architecture

Server-client (or client-server) architecture refers to a design paradigm where robots (clients) request services from a central server. In this architecture, the server hosts and manages the resources, services and information that the robots (clients) need in order to function. This server-client architecture provides the following benefits:

- Scalability: As the number of robots increases, more servers can simply be added or existing ones upgraded.
- Centralized management: It is easier to manage and maintain the robots as all updates, configurations and data management can be done centrally.
- Cost reduction: By offloading computation to the server, the robots can be built with less expensive hardware.
- Flexibility: It is easier to upgrade or modify the system as most changes can be done on the server side without having to physically modify the robots.

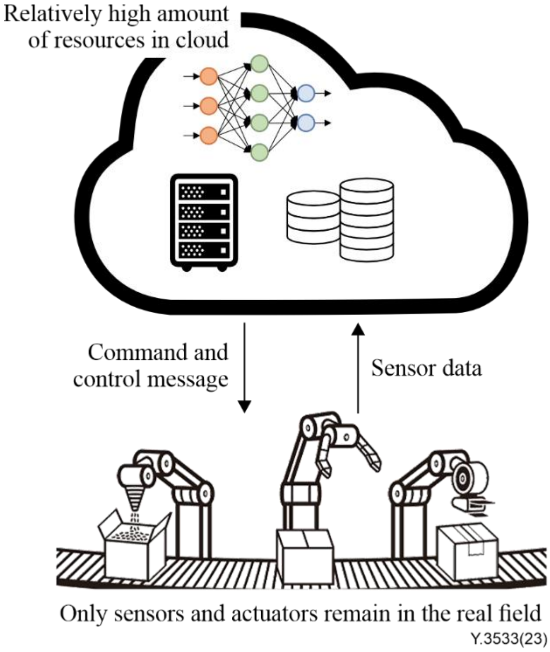
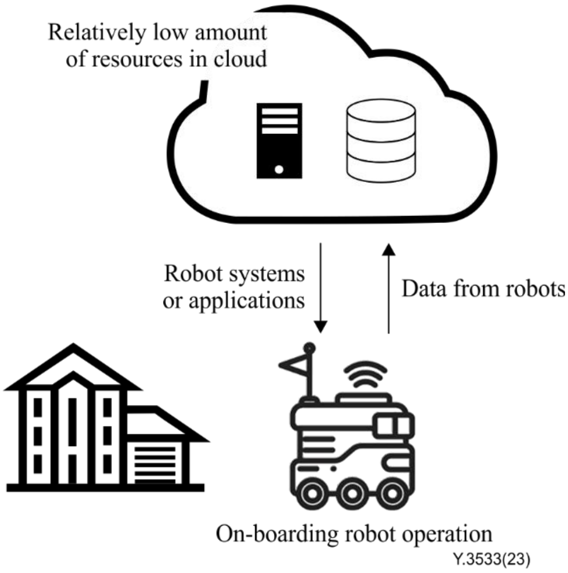
The server-client (robot) architecture is an efficient design paradigm that can enhance the scalability, manageability and cost-effectiveness of robot systems. However, it also comes with challenges, like ensuring reliable and fast network connections, handling server resources and managing security.

The deployment of robot systems with server-client architecture is flexible depending on the specific application and use case. Two conceptual deployment types of examples are provided for operations of robot systems.

In deployment example I, only sensors and actuators remain in the robots and other computational resources are offloaded to the server. All the applications and operations are processed in the server and the commands and control messages are transmitted to the robot via the network. The robot enables to perform complex tasks by leveraging cloud resources such as machine learning-based algorithms.

In deployment example II, the robot is physically located in any environment with enough computational resources to process on-board robot systems. The software and management system are updated over-the-air (OTA) which is served by the server, and the collected data which are sent for robot systems are stored in the server storage via the network.

Table III.1 – Deployment examples of server-client architecture of robotic system

	Deployment example I	Deployment example II
Physical deployment architecture		
Resources for robot systems	Computational resource for command and control is offloading in the cloud including data storage and operation system.	Data storage and management system are used.
Scalability changing for offloading	High demand for scalability changing to compute the robot task in cloud.	Less scalability changing is expected such as data storage for back-up storage
Network connection supports	The network latency affects the robot's performance. Even the physical location of the robot is restricted under the coverage of network.	Remote management such as monitoring of robot system is supported by network connection. The network capability only needs to update the data of robots or manage their on-board robot systems.

Bibliography

[b-ITU-T X.1601] Recommendation ITU-T X.1601 (2015), *Security framework for cloud computing.*

[b-ITU-T X.1631] Recommendation ITU-T X.1631 (2015) | ISO/IEC 27017:2015, *Information technology – Security techniques – Code of practice for information security controls based on ISO/IEC 27002 for cloud services.*

[b-ITU-T Y.2201] Recommendation ITU-T Y.2201 (2011), *Requirements and capabilities for ITU-T NGN*.

[b-ITU-T Y.2701] Recommendation ITU-T Y.2701 (2009), *Security requirements for NGN release 1*.

[b-ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014), *Information technology – Cloud computing – Overview and vocabulary*.

[b-ITU-T Y.3531] Recommendation ITU-T Y.3531 (2020), *Cloud computing - Functional requirements for machine learning as a service*.

[b-ISO 8373] International Standard ISO 8373:2021, *Robotics – Vocabulary.*

[b-ISO 10218-2] International Standard ISO 10218-2:2011, *Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration.*

[b-ISO 12100] International Standard ISO 12100:2010, *Safety of machinery – General principles for design – Risk assessment and risk reduction.*

[b-ISO 13482] International Standard ISO 13482:2014, *Robots and robotic devices – Safety requirements for personal care robots.*

[b-ISO 22166-1] International Standard ISO 22166-1:2021, *Robotics – Modularity for service robots – Part 1: General requirements.*

[b-ISO 22166-201] International Standard ISO 22166-201:2024, *Robotics – Modularity for service robots – Part 201: Common information model for modules.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2024