ITU-T

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.4453

(09/2016)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Internet of things and smart cities and communities – Frameworks, architectures and protocols

---

Adaptive software framework for Internet of things devices

Recommendation ITU-T Y.4453

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
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	
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	
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
Network control architectures and protocols	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.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
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.4453

Adaptive software framework for Internet of things devices

Summary

Recommendation ITU-T Y.4453 addresses the concept of the adaptive software framework (ASF), identifies high-level requirements and provides a reference functional architecture for Internet of things (IoT) devices. The ASF is a framework to manage and control adaptive IoT applications in real-time and in a dynamic way and enables optimal quality of service (QoS) performance. The adaptive IoT application is an IoT application that changes performance by altering system resource allocation (e.g., the number of central processing unit (CPU) cores, graphics processing unit (GPU) utilization, network bandwidth, etc.). Therefore, the ASF can provide optimal working of applications on IoT devices. The ASF is based on the IoT reference model described in Recommendation ITU-T Y.4000.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.4453	2016-09-13	20	11.1002/1000/13028

Keywords

Adaptive software framework, Internet of things, IoT.

---

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

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 received notice of intellectual property, protected by patents, 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 TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.4453

Adaptive software framework for Internet of things devices

1 Scope

This Recommendation describes the high-level requirements and functional architecture of the adaptive software framework (ASF) for Internet of things (IoT) devices.

In particular, the scope of this Recommendation includes:

– an overview of the ASF;

– features and high-level requirements of the ASF: monitoring capability, policy decision capability and management capability;

– functional architecture of the ASF: application monitoring manager function, system information manager function and policy manager function.

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.4000] Recommendation ITU-T Y.4000/Y.2060 (2012), *Overview of the Internet of things*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application [b-ITU-T Y.2091]: A structured set of capabilities, which provide value-added functionality supported by one or more services, which may be supported by an API interface.

3.1.2 Internet of things (IoT) [ITU-T Y.4000]: A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.

NOTE 1 – Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes full use of things to offer services to all kinds of applications, whilst ensuring that security and privacy requirements are fulfilled.

NOTE 2 – From a broader perspective, the IoT can be perceived as a vision with technological and societal implications.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 adaptive software framework: Middleware used to enable optimal QoS performance for each adaptive IoT application by using static/dynamic information regarding applications and system resources in IoT devices.

3.2.2 adaptive IoT application: A special IoT application working in the ASF. The application's performance is changed by altering system resource allocation (e.g., the number of CPU cores, GPU utilization, network bandwidth).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AP Access Point

API Application Programming Interface

ASF Adaptive Software Framework

CPU Central Processing Unit

DVFS Dynamic Voltage/Frequency Scaling

FE Functional Entity

GPU Graphics Processing Unit

ICT Information Communication Technology

IoT Internet of Things

IP Internet Protocol

IPC Inter-Process Communication

OS Operating System

PID Process Identification

QoS Quality of Service

STB Set-Top-Box

TCP Transmission Control Protocol

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 the adaptive software framework

With the rapid development of Internet of things (IoT) technologies, a large number of different IoT devices are entering the market. Usually, these IoT devices have varying hardware and software capabilities; therefore typical IoT applications need to be modified or customized to meet these diverse capabilities. It is necessary to manage and control these IoT applications to address this issue. The adaptive software framework (ASF) can manage and control adaptive IoT applications in real-time and in a dynamic way to enable optimal QoS performance. A conceptual diagram of the ASF consists of the IoT application layer, the adaptive software framework layer and the system resource layer, as shown in Figure 6-1. The ASF layer exists between the application layer and the system resource layer. The ASF performs monitoring for application and system monitoring, application and system management and analysis and policy decision-making.

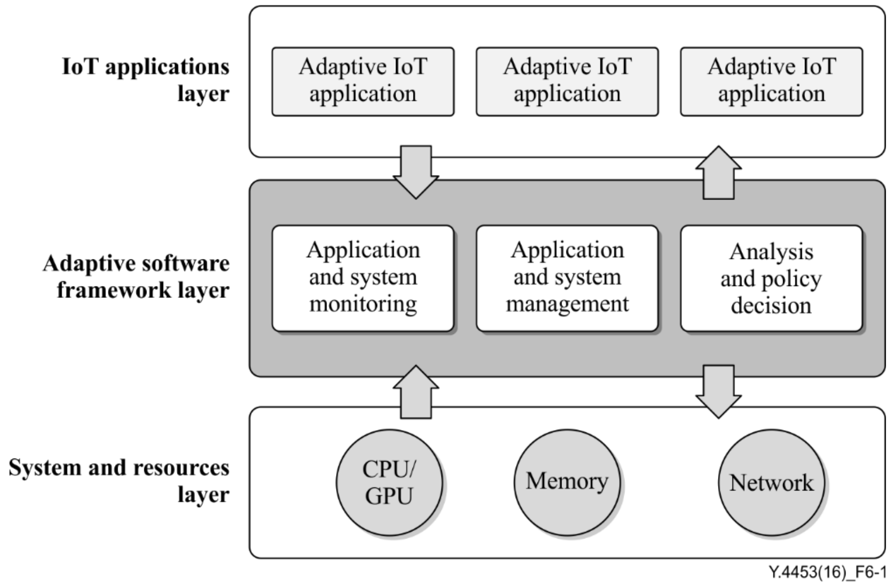

Figure 6-1 – A conceptual diagram of the ASF

The ASF gets more meaningful values under the current IoT service, because it enables resource-limited IoT devices to achieve high performance and compatibility over typical IoT systems. The ASF allows for the development of resource-managed IoT applications without a user's control. Also, the ASF provides functionalities to monitor and control application/system resources for IoT service interoperability. These functionalities make it easy to extend existing IoT systems.

The ASF enables developers to achieve high-performance, low power for applications. In addition, it can not only reduce typical software development complexity, but can also resolve the emerging complexities in diverse IoT applications and devices.

7 ASF features and high-level requirements

This clause describes the ASF features in accordance with the IoT reference model illustrated in Figure 7-1, as given by [ITU-T Y.4000] and high-level requirements of the ASF.

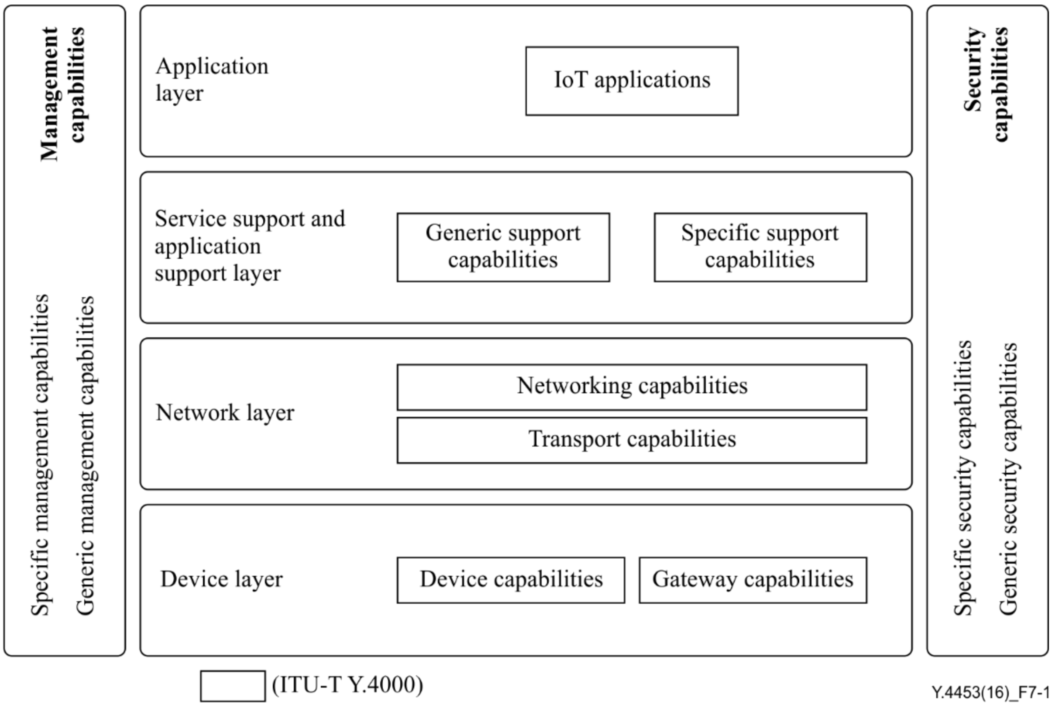

Figure 7-1 – IoT reference model per ITU-T Y.4000

7.1 ASF features

7.1.1 Application layer

The application layer considers adaptive IoT applications for the ASF.

– Internal adaptive IoT applications: The internal adaptive IoT application is an application that contains control codes.

– External adaptive IoT applications: The external adaptive IoT application is an application whose performance can be changed by the ASF.

7.1.2 Service support and application support layer

The service support and application support layer considers specific capabilities regarding application monitoring capabilities and policy managing capabilities for the ASF.

– Application monitoring capability: Monitors adaptive IoT applications in real-time by monitoring their QoS.

– Policy decision capability: Controls system resources such as CPU, memory and network status and decides adaptive IoT application's control by using static/dynamic information regarding applications and system resources.

7.1.3 Network layer

The ASF utilizes the network-related capabilities provided by the entities in the network layer.

7.1.4 Device layer

The device layer considers two kinds of devices for the ASF, small scale devices and large scale devices.

– Small scale devices: Terminal-node type of devices that are dependent on large scale devices, e.g., smart glasses.

– Large scale devices: Gateway devices and wired/wireless access point (AP) devices used in embedded and IoT devices, e.g., a set-top-box (STB).

7.1.5 Management capabilities

The management capabilities contain system information managing capability for the ASF.

– The system information managing capability has generic and specific management features. This collects and manages system information and application information. System information consists of static/dynamic data regarding CPU, memory and network status. Application information also consists of static/dynamic data regarding an application's user requirements and the application's QoS.
– The system information managing capability includes managing heterogeneous/homogeneous systems with single-core and multi-core processors. This can also have three managing features: management which receives the state of cores included in single-processor and/or multi-core processors of heterogeneous/homogeneous systems, determination which decides which core is to be allocated to execute an application based on the received states of the cores among these cores and allocation which allocates the determined core to execute the application.

7.1.6 Security capabilities

The security capabilities provide secure execution for the ASF.

– Security features include detecting and controlling execution requests of commands and services according to a security policy (e.g., a secure password is provided for the detected commands and services) and examining for the possibility of hacking behaviour for the execution of commands and services. If the commands and services are deemed secure as a result of a safety check of the commands and services, the secure command and services are executed and, if the commands and services are deemed unsafe, they are verified as secure through both a password check and a hacking behaviour examination executed in sequence.

7.2 High-level requirements

For development of adaptive IoT applications, it is recommended to:

– manage the development and execution environments of the typical applications;
– develop as a managed application distinctively for adaptive IoT application;
– separate the application's setting information and logic;
– provide a standard package format for application delivery.

7.2.1 Requirements for monitoring capability

For monitoring capability, it is recommended to:

– provide monitoring interfaces by internal and/or external method;
– manage monitoring (e.g., start and finish) by internal and/or external method;
– measure performance and/or resource utilization for instant and/or average sampling periods.

7.2.2 Requirements for policy decision capability

For policy decision capability, it is recommended to:

– provide parameter setup functionality for an internal adaptive IoT application to update an application's QoS;
– pre-define the priority of an application according to characteristics and provide the functionality to perform in sequence;
– support the dynamic setup for system resources (e.g., CPU, memory, network).

7.2.3 Requirements for management capability

For management capability, it is recommended to:

– provide mechanisms for collecting static and dynamic information about the system resources (e.g., CPU, GPU, memory, network) in real-time;

– support third-party modules that collect information (e.g., power, internal kernel status) from external modules and external devices.

8 Functional architecture of the ASF

The ASF can support heterogeneous/homogeneous types of embedded IoT devices and can include application/system manager functions (see Figure 8-1). These functions can adjust to allocate optimal system resources according to performance requirements of the adaptive IoT applications and system resources.

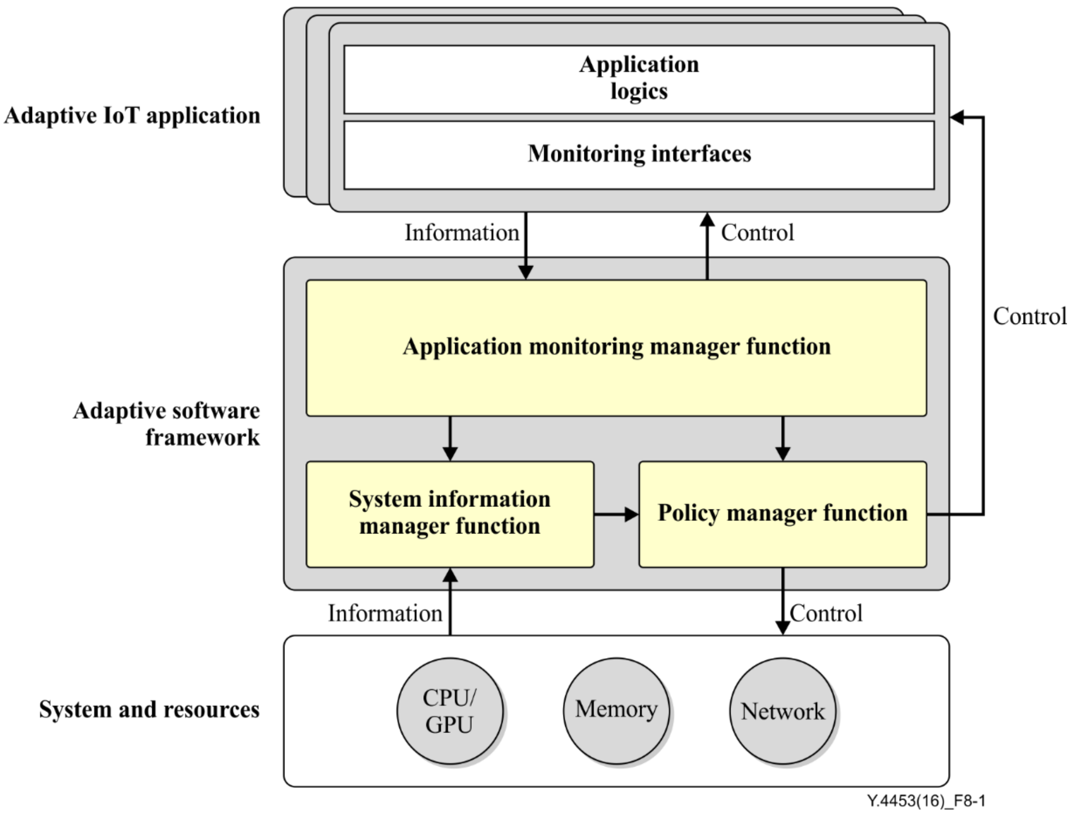

Figure 8-1 – Functional architecture of ASF

The ASF is composed of three functions:

1) the application monitoring manager function,

2) the system information manager function, and

3) the policy manager function.

The adaptive IoT application consists of application logic and monitoring capabilities providing information to the application monitoring manager function. The ASF gets information from the adaptive IoT application and the system and resources layers, and sends control signals (such as execution commands) to the systems and applications. The ASF optimizes computing performance for adaptive IoT applications.

8.1 Application monitoring manager function

The application monitoring manager function, shown in Figure 8-2, is the foundational part of the ASF. This function gets static and dynamic information from the adaptive IoT applications and then transfers this information in real-time to the system information manager function. Thus, if the system information manager function does not receive the adaptive IoT application's information, the ASF can't manage and control the adaptive IoT application. The application monitoring manager function receives an application's information through monitoring interfaces exposed by the adaptive IoT application and other information is generated by the application monitoring manager function. These two kinds of information are sent to the system information manager function using network protocols, e.g., inter-process communication (IPC) and/or transmission control protocol (TCP)/Internet protocol (IP). The application monitoring manager function is composed of three functional entities (FEs): application information gathering, QoS measurement and information processing.

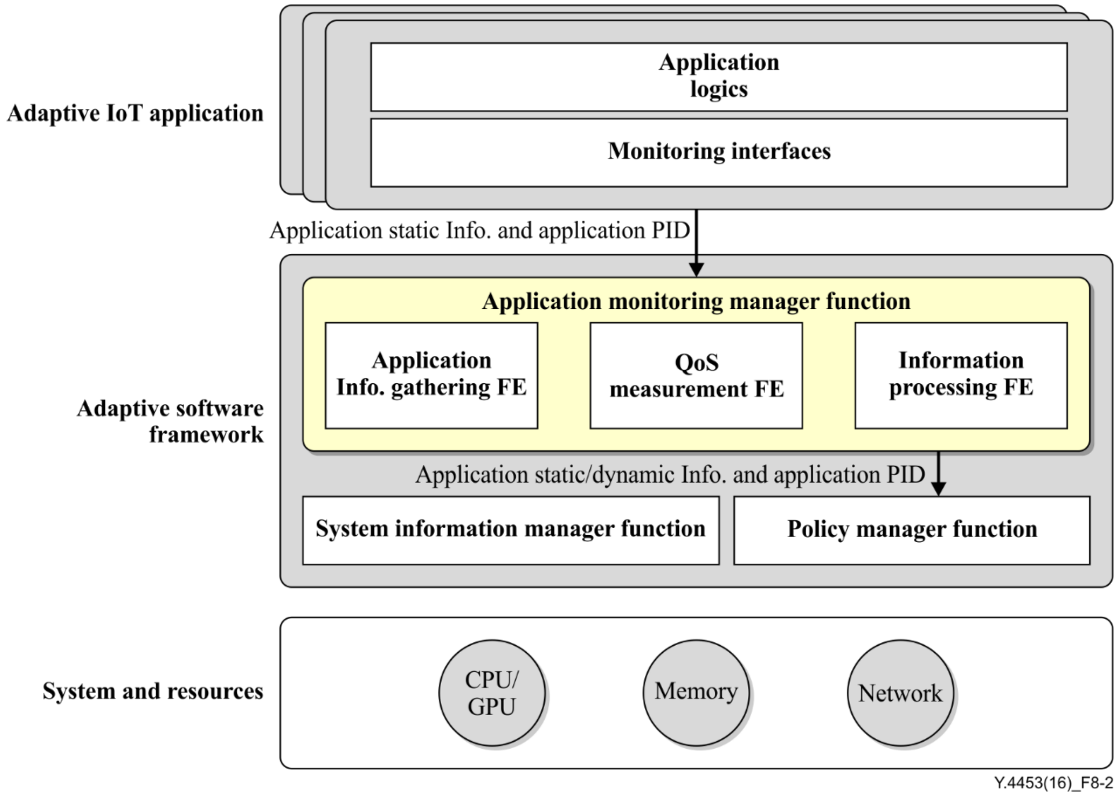

Figure 8-2 – Application monitoring manager function in ASF

8.1.1 Application information gathering FE

The application information gathering FE recognizes adaptive IoT applications by receiving their static information transferred through network protocols, e.g., IPC or TCP/IP. The following static information is necessary for adaptive IoT applications:

– max QoS value: maximum performance of application;

– min QoS value: minimum performance of application;

– target QoS value: performance desired by users;

– application process identification (PID): process identifier allocated by the system during the application's execution.

8.1.2 QoS measurement FE

The QoS measurement FE supports QoS performance. When adaptive IoT applications execute on systems, the QoS measurement FE generates a QoS performance rate. The QoS rate can be generated differently according to the current system state. Thus, in order to measure an application's performance, the QoS measurement generally provides three types of rates: global QoS rate, window QoS rate and instant QoS rate:

– global QoS rate: The entire average QoS rate calculated from the starting point and the ending point of the adaptive IoT application;
– window QoS rate: The average QoS rate calculated from the window size;
– instant QoS rate: The QoS rate calculated within a set time period.

For measuring QoS performance, the QoS measurement FE consists of five components: application registration which registers adaptive IoT applications, function explorer which detects user-defined functions in the application codes, loop-statement explorer which detects do-loop and for-loops in the application code, user-defined location explorer which detects locations (e.g., a line number) marked by a user in the application code and QoS generation which measures QoS performance corresponding to functions, loop-statements and user-defined locations detected in the application code.

8.1.3 Information processing FE

The information processing FE combines an application's static information such as its PID with the application's dynamic information such as its QoS; this combined information is then transferred to the system information manager function. According to a system's environment, the application monitoring manager function must provide a communication environment in which to send the static/dynamic information to the system information manager function. In order to provide transmission, the communication mechanisms are as follows:

– message queue: Message queues provide an asynchronous communications protocol, where the sender and receiver of the message do not need to interact with the message queue at the same time;
– shared memory: Shared memory is a very fast way of communicating. However, actions must be taken to avoid issues if processes sharing memory are running on separate CPUs and the underlying architecture is not cache coherent;
– sockets: Sockets provide point-to-point, two-way communication between two processes. Sockets are very versatile and are a basic component of inter-process and inter-system communications. A socket is a communication endpoint to which a name can be bound. It has a type and one or more associated processes.

8.1.4 Functionality for application monitoring manager function

The application monitoring manager function supports functionality to control adaptive IoT applications. Table 8-1 shows functionality for the application monitoring manager function.

Table 8-1 – Functionality for application monitoring manager function

Functionality	Description	Parameters	Return value
Requirement extraction of users and systems	An adaptive IoT application's PID is acquired. User and system requirements corresponding to the PID are extracted.	PID of a running application.	Character string including the requirement.

Table 8-1 – Functionality for application monitoring manager function

Functionality	Description	Parameters	Return value
Registration of adaptive IoT applications in the system information manager function	Adaptive IoT applications are registered into the system information manager function if a network connection is established.	Character string of information extracted in user and system requirements.	Message identifier generated to communicate between the adaptive IoT applications and the system information manager function
Transmission of adaptive IoT application's status information to the system information manager function	Status information generated in adaptive IoT applications is transferred to the system information manager function.	Process ID, Communication ID, QoS ID, Window QoS rate, Instant QoS rate, Global QoS rate, Timestamp (nanoseconds).	None
Adaptive IoT application receiving control result of the system information manager function	An adaptive IoT application receives a control result transmitting in the system information manager function.	QoS information.	Index of function point
Control of an adaptive IoT application	An adaptive IoT application performs a proper function by using received control results.	QoS information, Point of registered functions.	Return value regarding success or failure
Disconnection of an adaptive IoT application and the application monitoring manager function	An adaptive IoT application and the system information manager function cease network communication.	PID	None

8.2 System information manager function

The system information manager function, shown in Figure 8-3, gathers the system's static/dynamic information necessary for policy decision-making and transfers the relevant information to the policy manager function. It also includes a light-weight system information managing module for diverse IoT devices. This function includes the hardware's static information and can decide on the possible adaptive IoT application's execution and it allows the adaptive IoT application to decide compatibility and optimization.

The system information manager function is used to collect and manage system information in order to raise the execution efficiency of the adaptive IoT applications. The collected information is a criterion used to decide on the workflow of the adaptive IoT applications. In addition, the manager helps the applications to execute optimally by managing system information as well as process and QoS information.

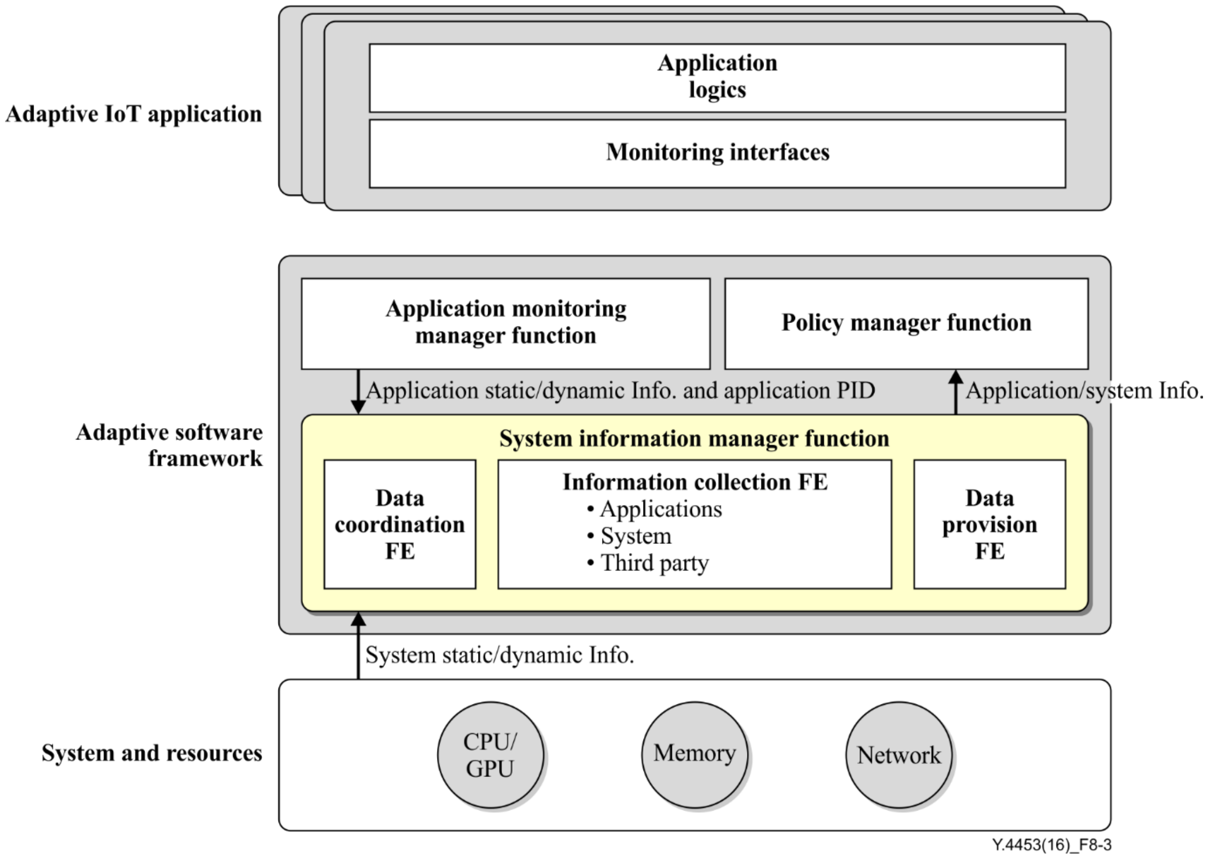

Figure 8-3 – System information manager function in ASF

The system information manager function consists of three FEs: data coordination FE, information collection FE and data provision FE. The system information manager function has two information inputs and one information output. The information is of two types: system static/dynamic information and application static/dynamic information. Static information is updated just once whenever the system information manager function is executed and dynamic information is updated periodically.

8.2.1 Data coordination FE

The data coordination FE reorganizes information acquired from systems (e.g., system static/dynamic information) and the application monitoring manager function (e.g., application static/dynamic information). Static information has one data structure corresponding to instant data. These data are classified as application, system and third-party type in real-time. Dynamic information is divided into two data structures: instant data and calculation data. The instant data (e.g., CPU activity) can be used in other modules immediately, while the calculation data (e.g., utilization) can be used with additional operations.

8.2.2 Information collection FE

The information collection FE gathers static and dynamic information in real-time for applications, systems and third parties. The system information also includes third party information collected by external modules or devices. System static information refers to fixed values regarding hardware resources (e.g., CPU, memory, network) and this information is set just once when a system begins. In general, system static information has a unique value in the system as long as hardware specific resources are not changed. This value is managed through one-time collection. System dynamic information refers to changeable values according to the system's status. Most of these values are different according to system overhead and they are updated periodically at every setup time for information collection.

8.2.2.1 System information

The system static information, shown in Table 8-2, is in regard to CPU core, CPU frequency, memory and network. The system static information is fixed information about the hardware. If the hardware capabilities are changed, system static information is also updated. This information includes the data structure corresponding to each hardware system. The system static information is updated once whenever the system information manager function is executed.

Table 8-2 – System static information

Information	Description
The number of cores	The number of CPU cores of a current system.
Max freq. of core	Maximum frequency by CPU core.
Min freq. of core	Minimum frequency by CPU core.
Core MIPS	Core performance presented with MIPS.
Core cluster	Core cluster in CPU environment (e.g., big.LITTLE Core).
Available freq.	The number of available frequency and frequency lists by core.
System memory	Memory usage of system.
Network interface	Network interface list to use in system.

The system dynamic information, shown in Table 8-3, is in regard to a CPU core's current state, CPU utilization, memory and network. The dynamic information can be changed according to the system's current state or users change system's resource values so that dynamic information is changed in real-time.

Table 8-3 – System dynamic information

Information	Description
Core activity status	Activity or inactivity of current core.
Core current freq.	Frequency configured in each core.
The number of threads	The number of threads to work on cores.
The number of processes	The number of processes to work on cores.
Core utilization	Core utilization allocated in current core.
Core time	Core usage time by core managing in OS.
System memory	Memory usage used in systems.
Network packet	Network Tx and Rx used in systems.

8.2.2.2 Application information

The ASF uses the application information to decide on and find optimal solutions. For supporting the framework, the information collection FE collects and manages information regarding adaptive IoT applications registered in the framework. The application information can be static or dynamic.

The adaptive IoT application's static information, shown in Table 8-4, is collected through the system information manager function. The static information consists of fixed information allocated from the OS during the adaptive IoT application's execution. This information corresponds to user requirements regarding an adaptive IoT application. The system information manager function can recognize the start and termination of an adaptive IoT application so that the function can find out exactly the period of time for which to get static information for the application. The information value manages QoS of adaptive IoT applications and the configuration information is used for tuning-up current QoS of the adaptive IoT applications. This information is set by users before the applications are executed and it is not changed until they are terminated.

Table 8-4 – Adaptive IoT application static information

Information	Description
Application program ID	Application program ID allocated by OS.
Program name	Execution name of application.
Program path	Absolute path of a running program.
Parent process ID	Parent ID for existing parent process.
Message queue ID	Message queue ID generated for information transmission.
Application process type	Application type of internal or external.
Max QoS	Maximum QoS available for application.
Min QoS	Minimum QoS available for application.
Target QoS	Target QoS to achieve in application.
Application window size	Window size in QoS framework.
Application sampling time	Sampling time to generate QoS.
Log file	File name of logging data.
Heartbeat buffer length	Buffer length of QoS record.

The adaptive IoT application's dynamic information, shown in Table 8-5, is collected through the system information manager function. The dynamic information is divided into information of the adaptive IoT application itself and the dynamic QoS of the adaptive IoT applications. The former information is updated by the OS or user configuration and the latter information is dependent on the application. The adaptive IoT application dynamic information consists of CPU, core, memory, network and QoS corresponding to the running applications.

Table 8-5 – Adaptive IoT application dynamic information

Information	Description
Allocated core	Core allocated in application program.
The number of threads	The number of threads allocated in application program.
Thread list	Thread ID list of application program.
Program status	Current status of application program.
Priority	Application program's priority allocated through OS.
Program time	Total time used in application program.
Program utilization	Utilization regarding core allocated in application program.
Memory utilization	Memory utilization used in application program.
Network information	Network usage used in application program.
QoS timestamp	Generation time of heartbeat in application program.
QoS window rate	QoS rate per time within window size allocated in application program.
QoS instant rate	QoS rate per time.
QoS global rate	QoS rate per time between start and current.

8.2.2.3 Third-party information

The third-party information, shown in Table 8-6, is provided by external modules or external devices. For example, a system's power is measured by an external device power-measuring system and the measured power-consumption data is transferred to the ASF. Besides these, other third-party information can be included whenever the information is needed by the ASF.

Table 8-6 – Example of third-party information

Information	Description
Power	Power consumption used in system.
Program characteristic	Program model analysed by program analyser.
Kernel inside information	Kernel information such as CPU scheduler and network stack provided by kernel module program.

8.2.3 Data provision FE

The data provision FE gets static/dynamic information regarding systems and applications from the information collection module. The data provision FE has communication facilities that efficiently transfer information gathered by the information collection module to the policy manager function. These facilities are implemented with various methods (e.g., message queue, shared memory, socket). The system information manager function should communicate with the policy manager function because the system information manager function plays an important role in deciding the optimal policy regarding adaptive IoT applications. The application/system information must be accurate in the aspect of time synchronization; this also provides information to the policy manager function for adaptive IoT applications.

8.2.4 Functionality of system information manager function

The resource monitoring interface transfers collected information to external modules, e.g., the policy manager, effectively by the system information manager. It includes two types of interfaces: system interfaces and application interface. Table 8-7 shows the functionality of the system information manager function.

Table 8-7 – Functionality of system information manager function

Functionality	Description	Parameters	Return value
Transmission of system static information	This functionality reads system static information and returns the information.	None	System static information.
Supply of system utilization	This functionality provides total system utilization or core utilization.	Core number	System utilization.
Supply of system frequency	This functionality provides core frequency.	Core number	Frequency.
Supply of application static information	This functionality reads application static information and returns the information.	PID	Application static information.
Supply of application utilization	This functionality provides total application utilization or core utilization.	PID	Application utilization.
Supply of application QoS	This functionality provides application QoS.	PID	Application QoS.

8.3 Policy manager function

The policy manager function, shown in Figure 8-4, provides the features of resource estimation, performance profiling for applications and systems and policy actuation. The policy manager function supports the application for optimal system resource allocation in IoT devices. It monitors the system state in real-time and controls optimal system resource allocation.

The policy manager function performs optimal utilization of system resources based on the requests of the adaptive IoT applications and system information manager function. Specifically, it analyses static and/or dynamic performance and power consumption in resources to guarantee high-performance and low power consumption through application monitoring information. It keeps the operational balance between the high-performance requirement and low-power consumption from the adaptive IoT application. Also, according to the application pattern, it predicts the amounts of power utilization and allocates the computing resource. In addition, it considers the information of available resources and details system performance information from the system information manager function.

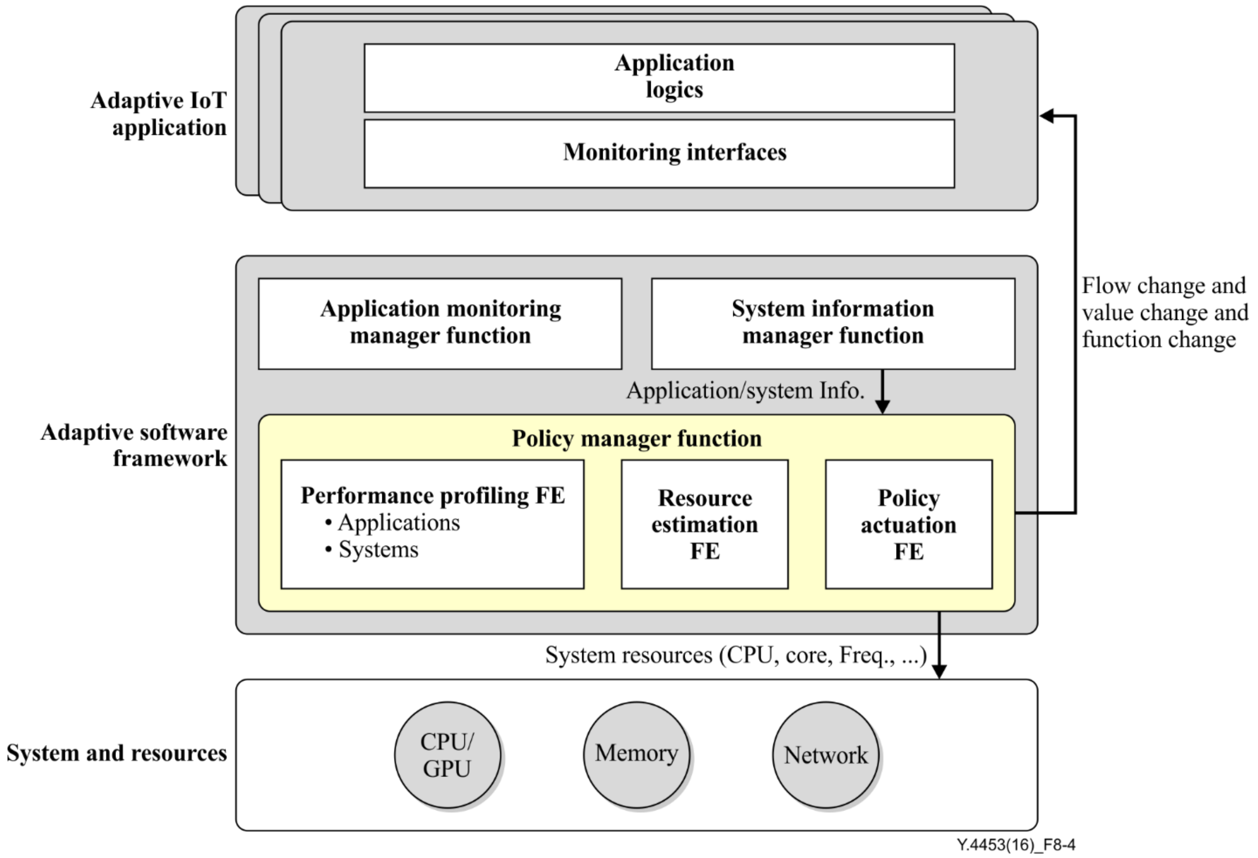

Figure 8-4 – Policy manager function in ASF

8.3.1 Performance profiling FE

The performance profiling FE provides the adaptive IoT application's optimization. The performance profiling is categorized into two types: adaptive IoT application profiling and system profiling. These two types of profiling are necessary, operating in parallel, to perform the adaptive IoT application's optimization. The two profiling types gather information regularly from the system information manager function.

NOTE – The information includes:

– application profiling information: PID, MAX QoS, MIN QoS, TARGET QoS, QoS rate, program ID;

– system profiling information: System total CPU throughput, CPU, memory, network packet rate (Tx, Rx), power consumption, PID information (core number, CPU frequency, memory, network packet Tx, Rx, power consumption).

In addition, the performance profiling FE detects various errors, such as race condition and energy bugs occurring in the adaptive IoT applications. It analyses these applications and provides optimal monitoring location candidates on their source code in order to measure QoS of the adaptive IoT applications.

– for detecting these errors, the performance profiling FE consists of two components: scalable race detection is configured to inspect the likelihood of a race of an access event with respect to a shared variable generated in a parallel program to select an access event that is likely to race. Power measuring is configured to measure power consumption when the parallel program is executed, to analyse whether an energy bug has been generated;

– for analysing monitoring location candidate, the performance profiling FE consists of four components: monitoring location analysis which detects monitoring location candidates in an input application ; monitoring range configuration which generates an analytical application using the application and the candidates; monitoring range analysis which detects monitoring range of each of the candidates through the analytical application; and monitoring environment configuration which determines which candidate has the greatest execution load (e.g., CPU utilization) as an optimal monitoring location for measuring QoS.

8.3.2 Resource estimation FE

The resource estimation FE predicts performance estimation for system resources and external estimation with external modules or third-party information. Performance estimation predicts CPU, memory and network performance. External estimation predicts power consumption from external modules. Resource estimation prediction occurs under the condition of no satisfaction regarding user requirement performance/power and system requirement performance/power.

8.3.3 Policy actuation FE

The policy actuation FE provides control for application and/or system resources through the resource estimation to guarantee the QoS performance and power.

The policy actuation FE controls application resources including: program logic flow control, program parameter change and program function change.

The policy actuation FE controls system resources including: CPU core on/off, core frequency, CPU dynamic voltage/frequency scaling (DVFS), memory swap and network bandwidth.

8.3.4 Functionality of policy manager function

This functionality supports the control of CPU, memory, network, etc. Table 8-8 shows the functionality of the policy manager function.

Table 8-8 – Functionality of policy manager function

Functionality	Description	Parameters	Return value
CPU resource control	Performs control operation regarding CPU core operation state, frequency according to user requirement	Core number, User requirement value	Success or Fail
Memory resource control	Performs memory utilization and size	Memory size, utilization rate	Success or Fail
Network resource control	Performs control operation regarding window size change according to user requirement	Running program's process ID, socket information	Success or Fail

Appendix I

An example use case and workflow for ASF

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

This appendix explains how the ASF manages and controls adaptive IoT applications. Figure I.1 shows an adaptive IoT application's transition of video quality, according to the application's resource policy and hardware resource conditions. The ASF is supporting, monitoring and controlling the adaptive IoT application's optimal performance. To guarantee the adaptive IoT application's optimal performance, the ASF works with the adaptive IoT application and IoT hardware resource information in real-time and in a dynamic way.

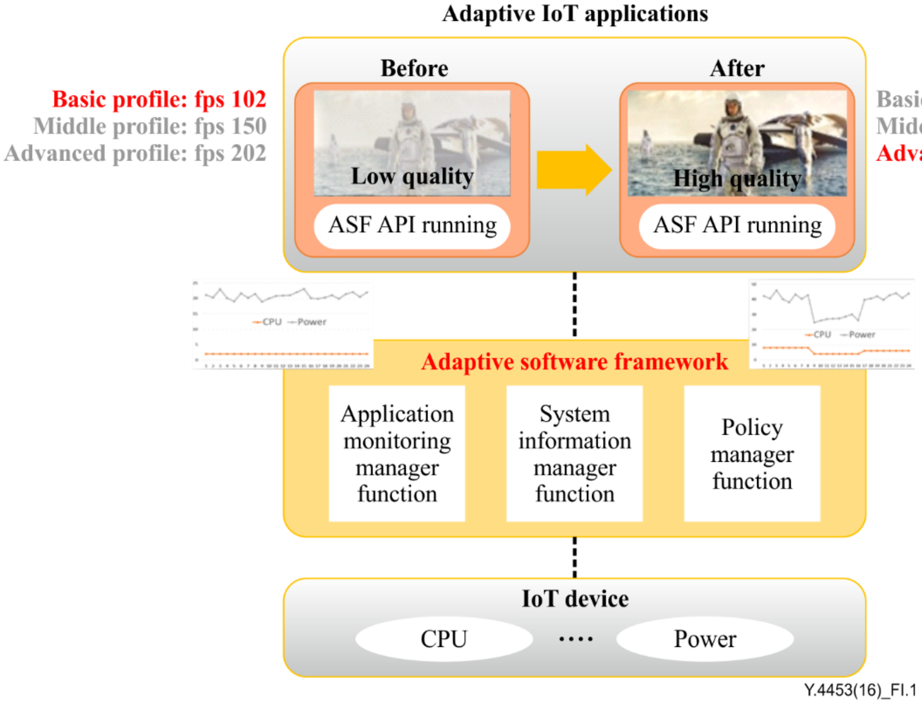

Figure I.1 – ASF for IoT devices

Figure I.2 shows a general model for an IoT service. The user is running an adaptive IoT application on a device and the device connects with the ASF. The IoT home gateway supports ASF functionality.

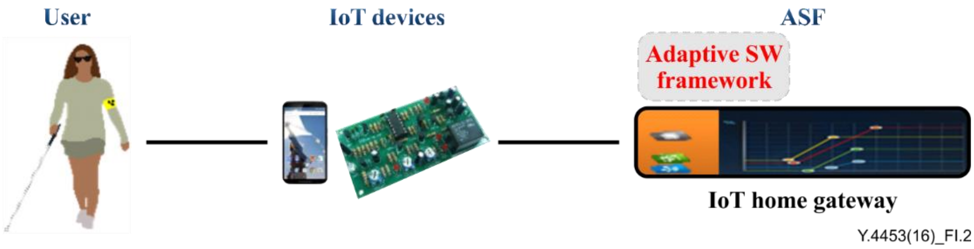

Figure I.2 - Conceptual model for IoT service

Figure I.3 shows when main ASF functionality is working and how the ASF is activated.

First, a user sends a service request to an adaptive IoT application and receives a video streaming service from the gateway. The gateway sends high quality video service to the user. However, if the

A/V QoS in the device is down, then the gateway detects heartbeat signals and the application module sends transfer heartbeat signals to the ASF engine in the gateway. Then the gateway transfers the higher-quality video streaming data to the IoT device according to the status information.

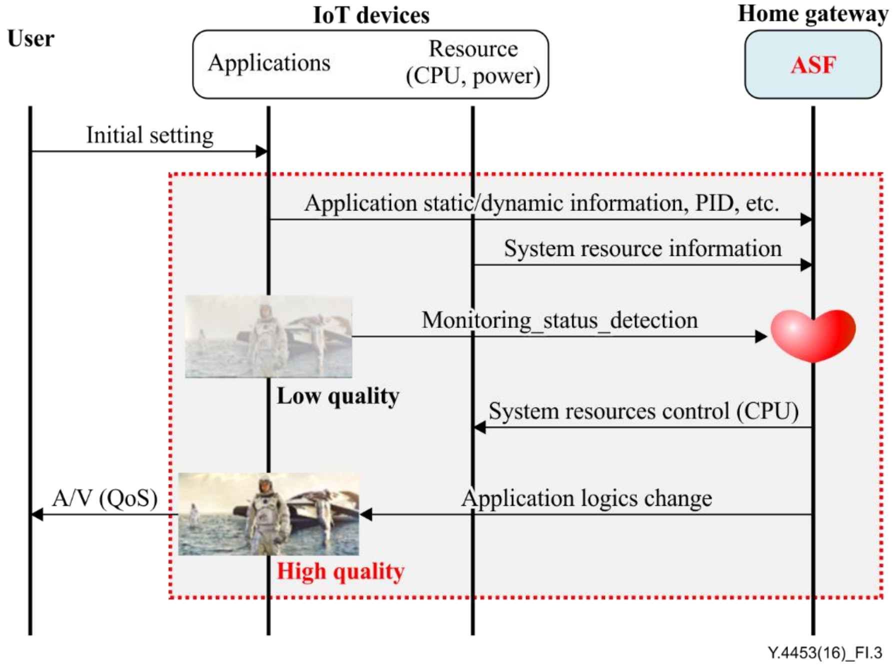

Figure I.3 – A procedure example of ASF

Bibliography

[b-ITU-T Y.2091] Recommendation ITU-T Y.2091 (2011), *Terms and definitions for next generation networks*.

[b-ITU-T Y.4414] Recommendation ITU-T Y.4414/H.623 (2015), *Web of things service architecture*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
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	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
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 and next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2016