International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1202

(01/2022)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Smart TV operating system

Smart television operating system – Architecture

Recommendation ITU-T J.1202

Recommendation ITU-T J.1202

Smart television operating system – Architecture

Summary

Recommendation ITU-T J.1202 specifies the architecture of a smart television operating system (TVOS) over integrated broadcast and broadband (IBB) cable networks. A smart TVOS is intended to be installed in an IBB-capable cable set top box (STB) and television (TV) to apply to broadcasting services and interactive services based on the Internet protocol provided by cable TV operators and third-parties. By running a smart TVOS, the IBB-capable STB and TV are able to intelligently provide subscribers with advanced and personalized services by downloading and installing advanced and personalized apps from the platforms of cable operators and third-parties, which are interconnected with them.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.1202	2019-07-29	9	11.1002/1000/13975
2.0	ITU-T J.1202	2022-01-13	9	11.1002/1000/14872

Keywords

Architecture, smart TV operating system, TVOS.

* 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 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

Introduction

This Recommendation is one of a series on a smart television operating system (TVOS). The Recommendations for this smart TVOS cover functional requirements, architecture, and security and application programming interfaces (APIs):

Smart television operating system – Functional requirements [b-ITU-T J.1201]

Smart television operating system – Architecture (ITU-T J.1202)

Smart television operating system – Specification [b-ITU-T J.1203]

Smart television operating system – Security framework [b-ITU-T J.1204]

Smart television operating system – Hardware abstract layer application programming interface [b-ITU-T J.1205]

Recommendation ITU-T J.1202

Smart television operating system – Architecture

1 Scope

This Recommendation specifies the architecture of a smart television operating system (TVOS) over integrated broadcast and broadband (IBB) cable networks. The smart TVOS is intended to enable an IBB-capable cable set top box (STB) and television (TV) to apply to broadcasting services and interactive services based on the Internet protocol provided by cable TV operators and third parties. By running the smart TVOS, the IBB capable STB and TV are able to intelligently provide subscribers with advanced and personalized services by downloading and installing advanced and personalized apps from the platforms of cable operators and third parties that are interconnected with them.

In this Recommendation, software components, frameworks, overall software architecture and security capabilities are also required by the smart TVOS.

The smart TVOS is not intended to support functionalities of the head-end systems of cable TV operators.

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.

None.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 integrated broadcast and broadband (IBB) DTV service [b-ITU-T J.205]: A service that simultaneously provides an integrated experience of broadcasting and interactivity relating to media content, data and applications from multiple sources, where the interactivity is sometimes associated with broadcasting programmes.

3.1.2 rich execution environment (REE) [b-ITU-T J.1201]: An extensible and versatile operating environment that brings flexibility and capability.

3.1.3 smart television operating system (TVOS) [b-ITU-T J.1201]: A system software running on integrated broadcast and broadband-capable (IBB-capable) cable set top box (STB) and television (TV) that are capable of managing hardware, software and data resources of the IBB-capable cable STB and TV, supporting and controlling the application software execution.

3.1.4 trusted execution environment (TEE) [b-ITU-T J.1201]: A secure area of the main processor in an integrated broadcast and broadband -capable cable set top box and television to ensure that sensitive data is stored, processed and protected in an isolated and trusted environment. It offers isolated safe execution of authorized security software providing end-to-end security by

enforcement of protected execution of authenticated code, confidentiality, authenticity, privacy, system integrity and data access rights.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 device tree source (DTS): A textual representation of a device tree in a form that can be processed and compiled into a binary device tree in the form expected by the operating system (OS). A device tree is a tree data structure with nodes that describe the physical devices in a hardware system. The tree data structure allows a runtime OS to run on top of the respective hardware system without hard-coding hardware information of the related devices included in the hardware system.

3.2.2 secure operating system: An operating system running in a trusted execution environment (TEE) which is used to trigger secure execution of applications within the TEE.

3.2.3 television operating system application framework: The software module of a smart television operating system that consists of application programming interface (API) units for constructing APIs with the computer programming language used by the respective applications and works along with the corresponding execution environment for the execution of respective applications.

3.2.4 television operating system execution environment (runtime): A software module in a smart television operating system (TVOS) that evaluates and executes applications consisting of computer language instructions, associated data and media content. An execution environment is implemented right above the component layer in a TVOS, and may be implemented along with computer language interpreters or language compilers, which an application may use to present audiovisual content, interact with a user or execute other tasks that are not evident to the user. A common example of an execution environment is that of Java software, using the Java programming language byte code interpreter and a Java virtual machine for program execution.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

AV Audio Video

DCAS Downloadable Conditional Access System

DRM Digital Rights Management

DTS Device Tree Source

DTV Digital Television

HAL Hardware Abstraction Layer

HCI Human-Computer Interaction

IBB Integrated Broadcast and Broadband

ID Identifier

IPC Inter-Process Communication

OS Operating System

OTT Over The Top

REE Rich Execution Environment

STB Set Top Box

TApp Trusted Application

TEE Trusted Execution Environment

TV Television

TVOS Television Operating System

5 Conventions

In this Recommendation:

The phrase "**is required to**" indicates a requirement that must be strictly followed and from which no deviation is permitted if conformity with this Recommendation is to be claimed.

The phrase "**is recommended**" indicates a requirement that is recommended but which is not absolutely required. Thus, this requirement needs not be present to claim conformity.

The phrase "**is prohibited from**" indicates a requirement that must be strictly followed and from which no deviation is permitted if conformity with this Recommendation is to be claimed.

The phrase "**can optionally**" indicates an optional requirement that is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator or service provider. Rather, it means the vendor may optionally provide the feature and still claim conformity with this Recommendation.

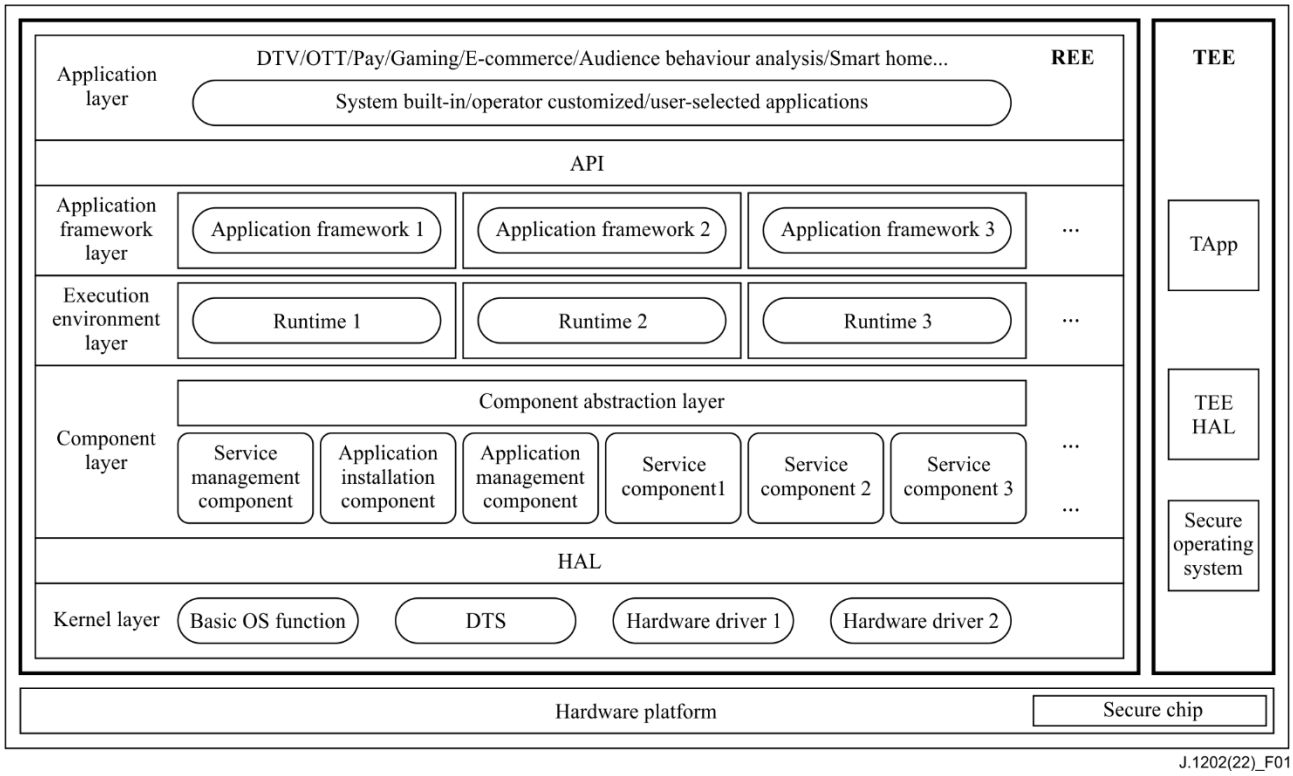
In this Recommendation the words *shall*, *shall not*, *should*, and *may* sometimes appear, in which case they are to be interpreted, respectively, as *is required to*, *is prohibited from*, *is recommended*, and *can optionally*. The appearance of such phrases or keywords in an appendix or in material explicitly marked as *informative* are to be interpreted as having no normative intent.

6 Functional architecture of the software

A TVOS consists of an REE and TEE.

The TVOS REE employs a hierarchical and modular software architecture and consists of five functional software layers: kernel; hardware abstraction layer (HAL); functional component; execution environment; and application framework in a loose-coupling mode. Each functional software layer consists of multiple software modules in a loose-coupling mode.

The TVOS TEE consists of the secure OS, TEE HAL and trusted application (TApp). Figure 1 shows the functional architecture of the TVOS software.



OTT: over the top

Figure 1 – Functional architecture of TVOS software

The TVOS kernel layer provides basic OS functions, including abstraction, management and allocation of system resources, such as process scheduling, memory management, virtual file system, network protocol stack, inter-process communication (IPC), security policies and hardware drivers. The kernel layer provides basic OS services for the upper-layer software.

The TVOS HAL provides abstraction and encapsulation of the TVOS hardware platform capability, uses a unified abstraction and encapsulation model for hardware devices of the same type, and offers unified invocation interfaces for the upper-layer software to access and control the hardware platform capability.

The TVOS functional component layer provides core functions of the TVOS and offers common service capability support for various applications. This layer contains shared functional component modules such as media processing, digital television (DTV), digital rights management (DRM), downloadable conditional access system (DCAS), secure payment, smart home, human-computer interaction (HCI), terminal control, application management and window management. Each shared functional component module is implemented in client-server mode and is independent from others. The server and client run in different process spaces and use the same mechanism to implement IPC. The server implements component functions and uses the HAL to invoke kernel-layer software modules and lower-layer hardware. The shared functional component modules can be added and tailored according to system requirements, and can support multiple execution environments.

The TVOS execution environment layer implements the interpretative execution environment of the application software and application adaptation software, and supports the loading and running of multiple types of applications such as those of Java and the web. The TVOS execution environment layer supports can be added and tailored according to system requirements. Each TVOS execution environment is independent from others and has its own application framework. For example, the execution environment includes runtime for the web, Java or Python.

The TVOS application framework layer provides APIs to different types of applications. Each TVOS application framework is independent from others and corresponds to one execution environment.

7 Kernel layer

The TVOS kernel layer contains a kernel, DTS and hardware driver modules such as Linux.

The kernel implements basic OS functions such as process scheduling, memory management, VFS, network protocol stack, input/output management, IPC and security protection. The kernel collaborates with a secure chip to support and implement the security trust chain verification mechanism based on the hardware security trust root.

The DTS software module should decouple the kernel from hardware drivers based on the DTS mechanism.

The hardware driver software module should contain various universal hardware drivers, DTV-related hardware drivers and security-related hardware drivers, and support the TVOS kernel to communicate with, access and manage various hardware devices.

8 Hardware abstraction layer

The TVOS HAL consists of multiple hardware abstraction functional interface modules. These modules implement abstraction and encapsulation of different hardware capabilities, and provide the upper-layer software with interfaces used to invoke the corresponding hardware capabilities.

The hardware abstraction functional interface modules are implemented by using the stub hardware abstraction model, which takes a hardware module, hardware devices and their operation methods as a stub operation function. By mapping the hardware module identifier (ID) to the corresponding pointer of the stub operation function, the stub hardware abstraction model provides upper-layer software with the invocation method of the hardware capability, implementing operation and control of hardware capabilities.

Figure 2 shows the principle of the stub hardware abstraction model of the TVOS HAL.

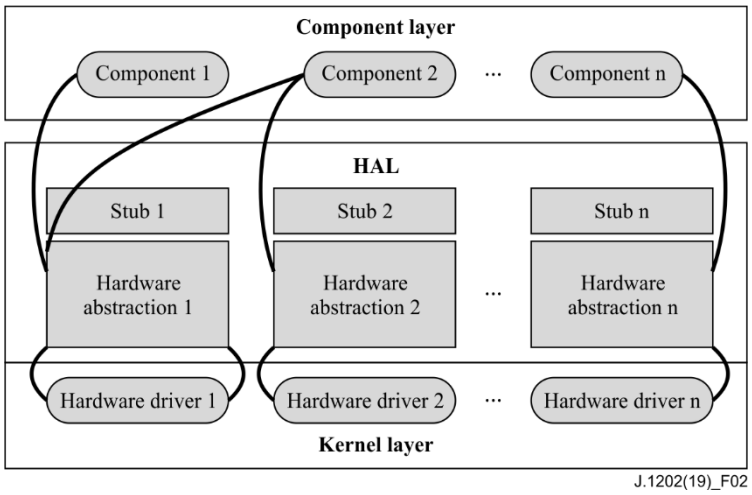


Figure 2 – Principle of the stub hardware abstraction model of the TVOS HAL

The TVOS HAL functional interface modules should include the HAL functional interface modules for the hardware dedicated to media processing and generic hardware.

The HAL functional interfaces for the hardware dedicated to media processing belong to hardware abstraction interfaces.

9 Component layer

9.1 Component model

The TVOS component consists of the server and client, which run in different process spaces and use the binder mechanism to implement IPC. The server is responsible for implementing component functions and uses the HAL to invoke kernel-layer software modules and lower-layer hardware. The component server is a system resident running instance and contains software modules including functional implementation of a related service and a service stub that is an abstract implementation of the corresponding service and works as the IPC interface for exporting the service to the client. A component server running instance provides services for multiple different component client running instances. The component client contains software modules including client implementation, client APIs and a service proxy. A service proxy sends the requests being executed in the remote service process to the server and gets the corresponding reply to the client. The server and client of the shared functional component modules should be implemented by using an appropriate programming language to work with various execution environments as illustrated in Figure 3.

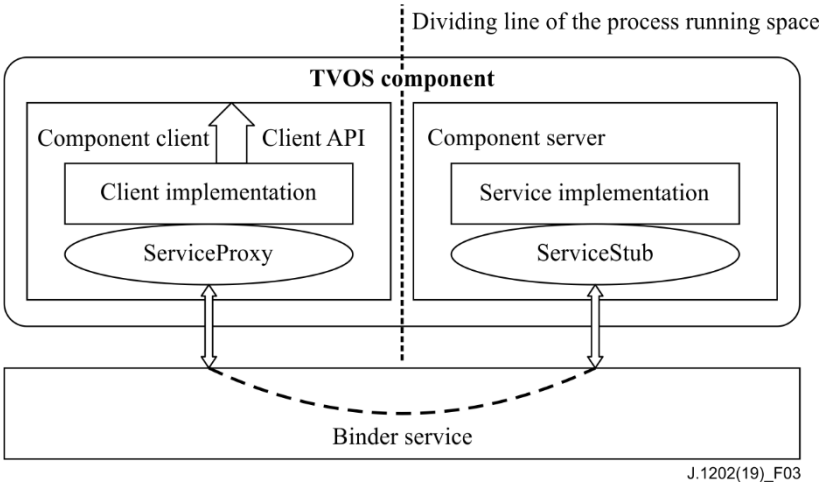


Figure 3 – Component model diagram

The collaboration between the component server and client requires support from the component service manager.

The component server registers the server information with the component service manager. The component client queries information about the corresponding component server from the component service manager to invoke the component server.

The component service manager parses the component names, maintains the mapping between the component names and component instances, and checks and controls the access rights of the component server. The principle of collaboration between the component and the component service manager is shown in Figure 4.

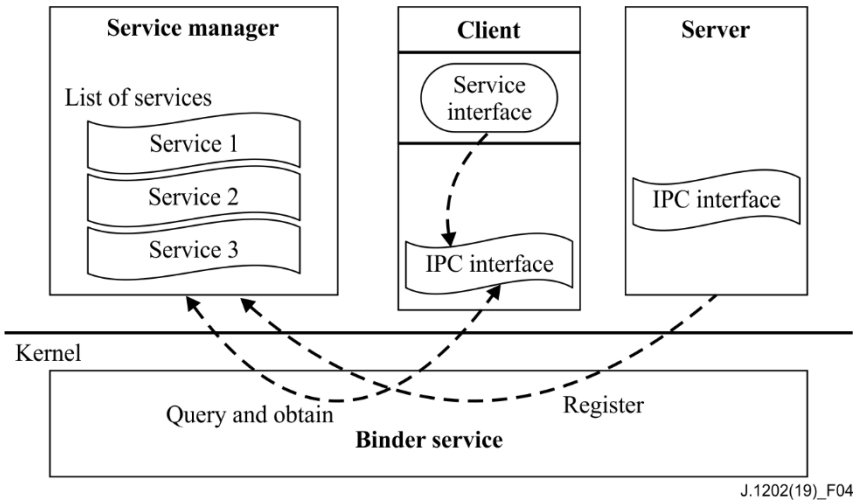


Figure 4 – Principle of collaboration between the component and component service manager

9.2 Component service manager

9.2.1 Functions

The component service manager should implement the following functions:

– manage all components in the system in a centralized manner and provide the function of registering TVOS components;

– register a component with checking whether a component is effectively registered, including confirming whether the component is legal, whether the component is repeatedly registered, and whether the component can be allocated with sufficient resources;

– provide other software modules and applications with the function of searching for components and obtaining component clients.

The component service manager is a special component that provides the component management function and complies with the TVOS component model.

9.2.2 Component implementation and invocation mode

The TVOS component service manager should be implemented according to the component model. Figure 5 shows the component implementation and invocation mode.

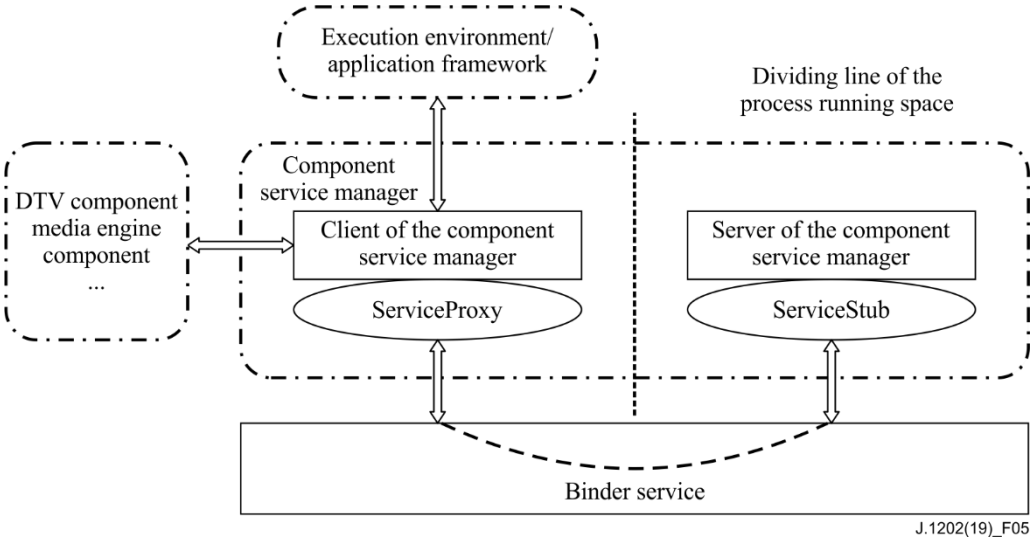


Figure 5 – Implementation and invocation mode of the component service manager

9.2.3 Functional architecture and modules

The component service manager consists of component registration, component validity check and component query modules, which correspond to the itemized bullets in clause 9.2.1. Figure 6 shows the architecture of the component service manager.

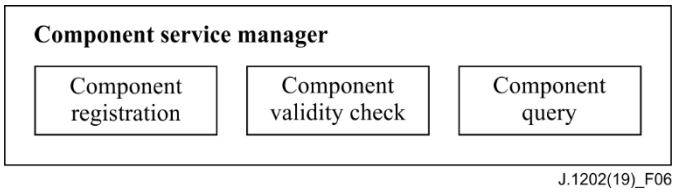


Figure 6 – Functions architecture and modules of the component service manager

The component registration module responds to requests to register other components with the component service manager.

The component validity check module verifies whether; the component permissions are legal; a component is repeatedly registered; and resources are sufficient to implement registration.

The component query module queries components and obtains component clients.

9.2.4 Interfaces

The component service manager should provide other software modules with component registration interfaces and applications with component query and component client-obtaining interfaces by using the client. The component service manager provides four functional interfaces for services, namely adding, checking, obtaining and listing.

9.2.5 Role of component service manager

The component service manager is a special component to register and query all other components and applications. The component service manager relies on the system lower-layer binder service to manage other components.

9.3 Application installation component

9.3.1 Functions

The application installation component should implement the following functions:

– parse TVOS application packets;

– authenticate the security of TVOS application packets, including integrity and validity authentication and permission check;

– install, uninstall and update TVOS applications;

– query the installation information of TVOS applications;

– support the application management component to manage installed applications.

9.3.2 Component implementation and invocation mode

The TVOS application installation component should be implemented according to the component model. Figure 7 shows the component implementation and invocation mode.

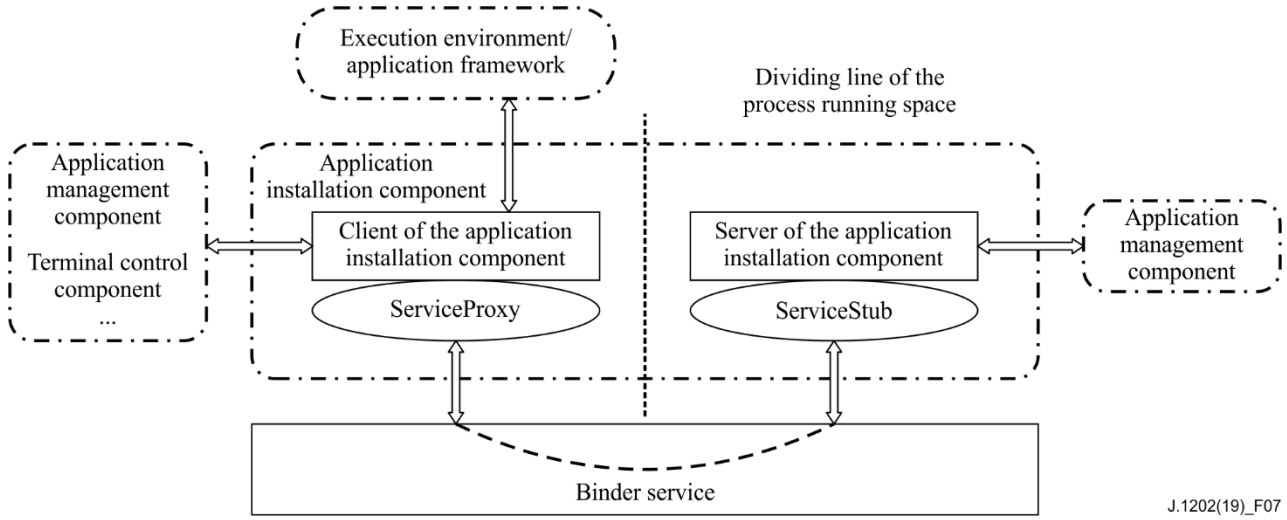


Figure 7 – Implementation and invocation mode of the TVOS application installation component

9.3.3 Functional architecture and modules

The application installation component consists of modules for parsing, installation, security and query for an application. Figure 8 shows the architecture of the application installation component.

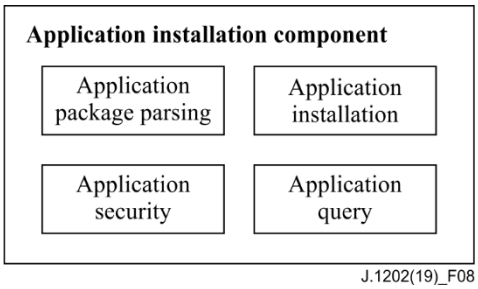


Figure 8 – Architecture and modules of the application installation component

The application package parsing module parses the application package to be installed and checks the installation space required for corresponding applications.

The application security module verifies the integrity and validity of the application to be installed and checks the permission requests of the application.

The application installation module installs, uninstalls and updates applications.

The application query module allows other software modules and applications to query information about the applications installed.

9.3.4 Interfaces

The application installation component should provide other software modules with invocation interfaces used to install, uninstall, and update TVOS applications and interfaces used to query the installation information of the installed TVOS applications. These interfaces are functional component interfaces.

9.3.5 Collaboration with other software modules

The application installation component and the application management component are interdependent. The application installation component collaborates with the application management component to implement related functions and also provides support for the application management component. The application installation component also provides the information query service for other software modules and applications.

9.4 Application management component

9.4.1 Functions

The application management component should implement the following functions:

– registration and deregistration of TVOS applications;

– management of the life cycle of the running states (including start, stop, pause, resume and exit) of TVOS applications;

– implementation of the communication mechanism between different software modules and applications, including the messaging mechanism for both single and multiple targets;

– implementation of the data-sharing mechanism between different software modules and applications, as well as information about the method of data sharing, and a mechanism to notify data changes;

– assistance to the application installation component for installing and updating applications.

9.4.2 Component implementation and invocation mode

The application management component should be implemented according to the component model. Figure 9 shows the component implementation and invocation mode.

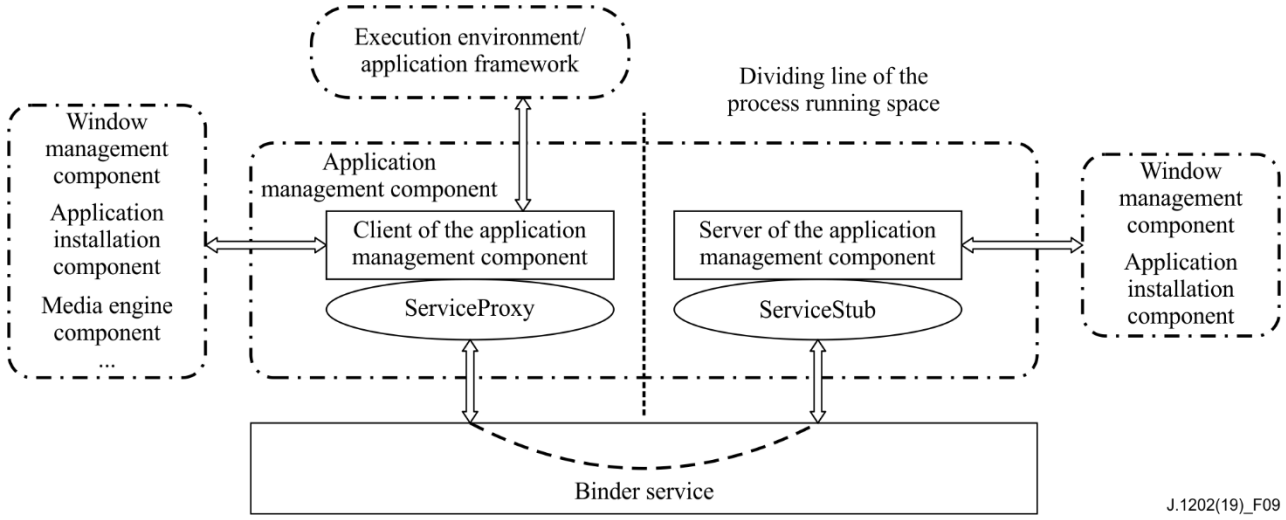


Figure 9 – Implementation and invocation mode of the application management component

9.4.3 Functional architecture and modules

The application management component contains modules to manage application memory, process, application scheduling, application life cycle, messaging and data sharing. Figure 10 shows the functional architecture and modules of the application management component.

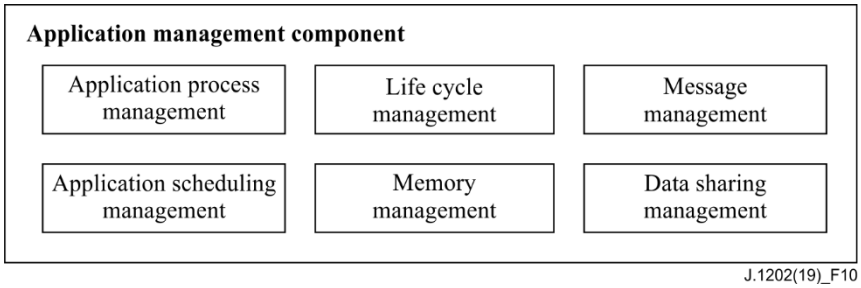


Figure 10 – Functional architecture and modules of the application management component

The application process management module manages for the application registration and deregistration, allocation of a unique ID and allocation of resources.

The application memory management module manages the allocation of memory for the application.

The application life cycle management module manages the application running states such as start, running, suspend, resume and stop.

The application scheduling management module schedules and manages the state of the applications considered, including activation, suspension, and switching the execution between foreground and background.

The messaging management module provides the messaging mechanism for both single and multiple targets for communication between different software modules and applications.

The data-sharing management module implements the data-sharing mechanism to support and manage data sharing between different software modules and applications.

9.4.4 Interfaces

The application management component should provide interfaces used to control start and stop of applications, as well as interfaces used to query information about running applications through the client.

9.4.5 Collaboration with other software modules

Installation, uninstallation and execution of the applications are achieved by collaboration of components for application management, application installation and window management. The application management component and application installation component are key to application installation and uninstallation. The application management component and window management component are key to application execution. For that purpose, the application management component provides support for the implementation and running of other components and applications.

9.5 Service components

9.5.1 Functions

A TVOS may implement various services by different service components. A TVOS should support the following:

– DTV component for DTV-related service;

– media engine component for media service;

– DCAS component for DCAS service;

– DRM component for DRM service;

– HCI component for human-computer interface service.

9.5.2 Component implementation and invocation mode

TVOS service components should be implemented according to the component model. Figure 11 shows the component implementation and invocation mode.

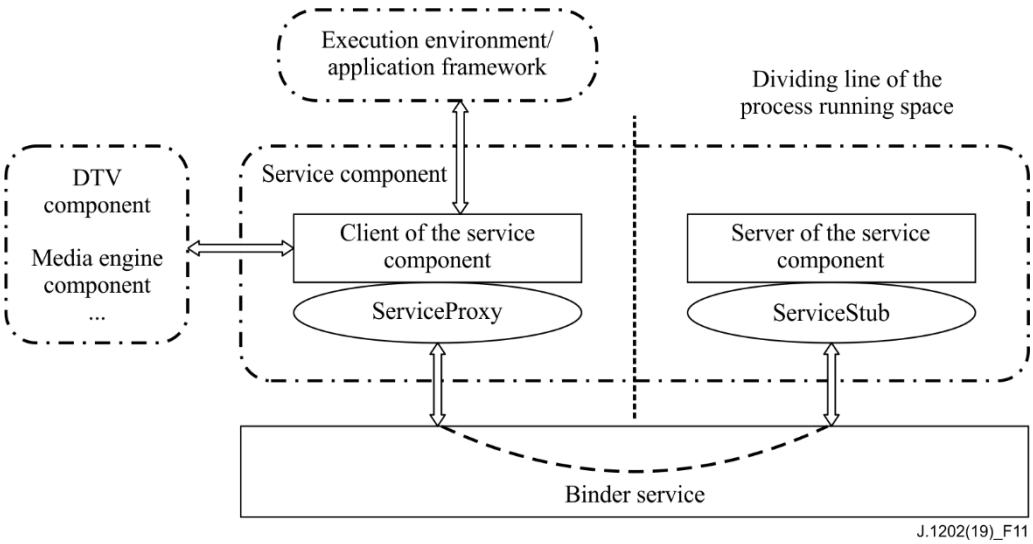


Figure 11 – Implementation and invocation mode of the service component

10 Application execution environment

10.1 Functions

TVOS execution environments should implement the following functions:

– the interpretation and running environment for TVOS applications and the application framework-layer functional interface instances invoked by TVOS applications;

– loading and running of TVOS applications;

– process isolation for TVOS applications by use of a different runtime instance for each TVOS application;

– cooperation with the application management component to manage the life cycle of TVOS applications, including their starting, pausing, resuming and restarting;

– managing the access rights of TVOS application resources, including checking and requesting the rights of TVOS applications.

10.2 Architecture and implementation mechanism

TVOS execution environments create and provide the running environment for TVOS applications and manage their running, rights and security.

Figure 12 shows the architecture and implementation of TVOS execution environment.

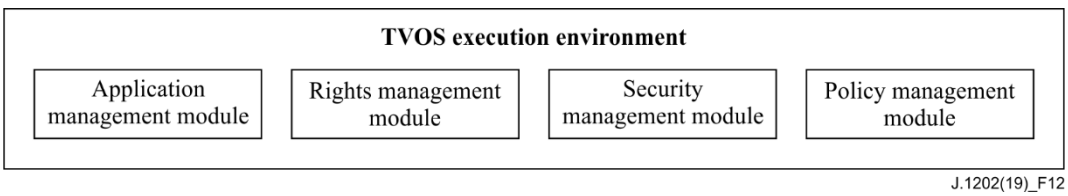


Figure 12 – Architecture and implementation of a TVOS execution environment

A TVOS execution environment consists of management modules for application running, rights and security.

The application management module creates instances of a TVOS execution environment, forms the basic environment of TVOS applications, loads applications to the basic environment, and starts and manages the life cycle of applications.

The rights management module manages the access rights of TVOS application resources when TVOS applications are running.

The security management module implementation includes process isolation and data isolation.

The policy management module implementation for the running modes of TVOS applications includes the policy to exclusively occupy processes for each application or share the same process for multiple applications.

11 Application framework

11.1 Architecture of the TVOS application framework

The TVOS application framework consists of the TVOS application programming interface units and corresponding interpretative execution environment functional interface units.

11.2 TVOS application programming interface units

11.2.1 Functions

The TVOS application programming interface units implement interfaces of various functional components and modules, provide application programming interfaces, and assist applications in implementing DTV services such as electronic programme guide (EPG), channel list and TV programme playing.

The TVOS application programming interface units include:

– unidirectional broadcast network access unit;

– broadcast protocol processing unit;

– bidirectional broadband access unit;

– HCI unit;

– audio video (AV) setting unit;

– media processing unit;

– message management unit;

– broadcast information service unit.

11.2.2 Main functional interface units

The main functional interface units are as follows.

a) Unidirectional broadcast functional interface unit

This unit cooperates with the DTV component to access the unidirectional cable DTV broadcast network, including controlling the frequency, modulation mode and symbol rate as well as obtaining information such as the signal strength and quality. This unit also cooperates with the DTV component to implement basic DTV functions such as programme search, programme guide and information search, supporting the running of DTV-related applications.

b) Bidirectional broadband network functional interface unit

This unit cooperates with the network service to implement bidirectional broadband network access, including connection management.

c) HCI functional interface unit

This unit cooperates with the HCI component to implement the user interface. The user instructions can be obtained by input devices such as the remote control, mouse, keyboard

and front-panel key in the form of key messages. The information to the user is shown to the front panel or display screen.

d) Configuration functional interface unit

This unit cooperates with the AV configuration component of some applications to configure audio and video output parameters, including global volume, volume status of the audio output, brightness, contrast and saturation of the video output.

e) Media playing functional interface unit

This unit cooperates with the media component to implement media playing functions, including playing, control, language selection, event processing and exception processing.

f) Message management functional interface unit

This unit is the TVOS message repository for applications dealing with a message event, message event listener, and message manager. This unit manages and distributes messages and assists applications in message acquisition.

g) DCAS functional interface unit

This unit cooperates with the DCAS component to implement DCAS application management, entitlement control message/entitlement management message data processing, and DCAS data interaction, assisting DTV applications in playing encrypted DCAS programmes. Some interfaces may have access limited by the application.

h) Terminal control data collection unit

This unit cooperates with the terminal control component and other components for some applications to interact with the terminal control and data collection components.

i) Broadcast message service unit

This unit provides functional interfaces for monitoring, reception and processing broadcast information services.

11.2.3 Relationship of the interfaces and functional components

The framework-layer TVOS application programming interface depends on function support from components in the component layer. Figure 13 shows the interface relationship.

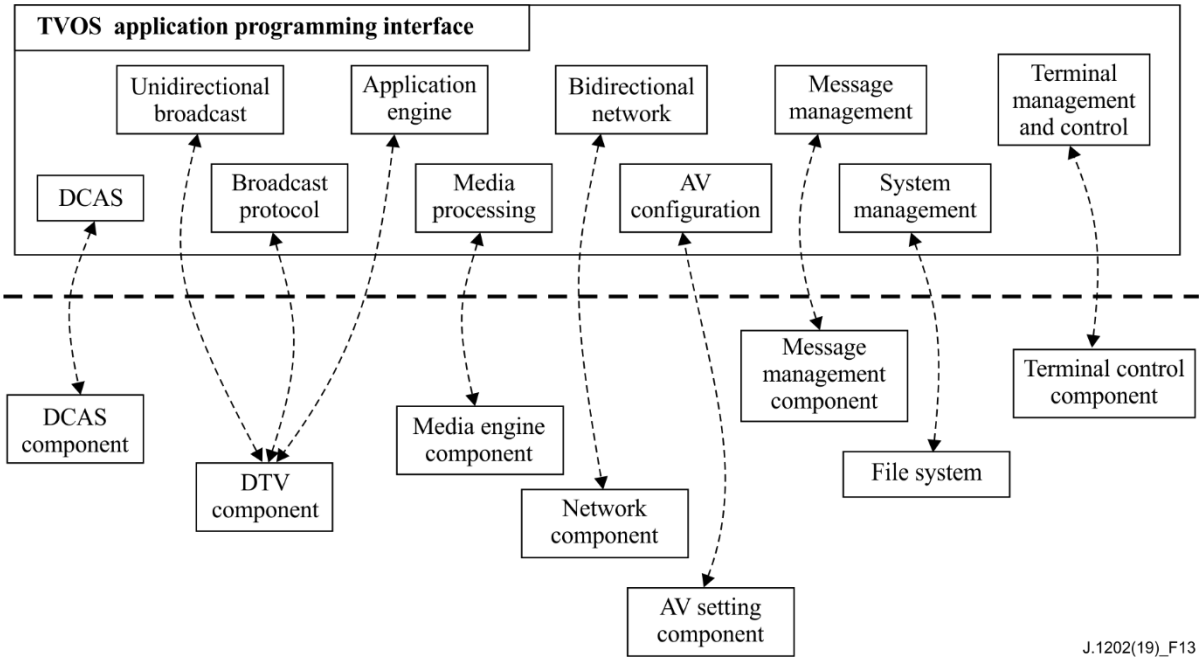


Figure 13 – TVOS application programming interface of the framework layer

11.3 Interpretative execution environment functional interface units

Interpretative execution environment functional interface units support the standard application programming interfaces of each interpretative programming language.

Bibliography

[b-ITU-T J.205] Recommendation ITU-T J.205 (2012), *Requirements for an application control framework using integrated broadcast and broadband digital television*.

[b-ITU-T J.1201] Recommendation ITU-T ITU-T J.1201 (2022), *Smart TV operating system – Functional requirements*.

[b-ITU-T J.1203] Recommendation ITU-T ITU-T J.1203 (2022), *Smart television operating system – Specification.*

[b-ITU-T J.1204] Recommendation ITU-T ITU-T J.1204 (2022), *Smart television operating system – Security framework.*

[b-ITU-T J.1205] Recommendation ITU-T ITU-T J.1205 (2022), *Smart television operating system – Hardware abstract layer application program interface.*

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

Printed in Switzerland
Geneva, 2022