International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.765

(04/2015)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

IPTV multimedia services and applications for IPTV –
IPTV multimedia application frameworks

Packaged IPTV application (widget) service

Recommendation ITU-T H.765

ITU-T

International
Telecommunication
Union

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Telepresence	H.420–H.429
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
Mobility interworking procedures	H.550–H.559
Mobile multimedia collaboration inter-working procedures	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	H.620–H.629
Ubiquitous sensor network applications and Internet of Things	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	**H.760–H.769**
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789
E-HEALTH MULTIMEDIA SERVICES AND APPLICATIONS	
Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)	H.820–H.859
Multimedia e-health data exchange services	H.860–H.869

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

Recommendation ITU-T H.765

Packaged IPTV application (widget) service

Summary

Packaged internet protocol television (IPTV) applications (IPTV widgets) are lightweight applications, provided with a simple and easily accessible graphical user interface (GUI), that commonly provide a single functionality (e.g., calendars, feed aggregators, etc.) and might stay persistently exhibited.

IPTV widgets may run on different types of IPTV terminal devices and therefore standardized technologies must be supported by the widget engine in the terminal device. Recommendation ITU-T H.765 describes IPTV widget services that are developed using the technologies defined in the ITU-T H.760 multimedia application framework (MAFR) series, such as nested context language (NCL) (Recommendation ITU-T H.761), LIME (Recommendation ITU-T H.762), HTML (new MAFR ITU-T Recommendation under study), cascading style sheets (CSS) (Recommendation ITU-T H.763.1), Services enhanced scripting language (SESL) (Recommendation ITU-T H.764) and Lua (new MAFR ITU-T Recommendation under study). The widget service also takes advantage of functionalities described in the ITU-T H.721 IPTV terminal device basic model, as well as service discovery protocols and services specified in Recommendation ITU-T H.770.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.765	2015-04-29	16	11.1002/1000/12464

[*] 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, 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

Page

Recommendation ITU-T H.765

Packaged IPTV application (widget) service

1 Scope

This Recommendation specifies the architecture and functional components needed for the provisioning of packaged IPTV application (widget) service. Widgets are lightweight applications, provided with a simple and easily accessible GUI, which commonly provide a single functionality (e.g., calendars, feed aggregators, etc.) and might stay persistently exhibited. The widget service definition relies on the interoperability and harmonization provided by IPTV multimedia application frameworks (MAFRs) [ITU-T H.760]. The scope of the recommendation comprises the definition of the IPTV packaged application model and the definition of essential service components such as service discovery, packaging, metadata and security.

The target terminal device types range from the basic model as specified in [ITU-T H.721] to full-fledged terminal devices, which are defined in [ITU-T H.722].

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 H.721] Recommendation ITU-T H.721 (2015), *IPTV terminal devices: Basic model*.

[ITU-T H.722] Recommendation ITU-T H.722 (2014), *IPTV terminal devices: Full-fledged model*.

[ITU-T H.760] Recommendation ITU-T H.760 (2009), *Overview of multimedia application frameworks for IPTV services*.

[ITU-T H.761] Recommendation ITU-T H.761 (2014), *Nested context language (NCL) and Ginga-NCL*.

[ITU-T H.762] Recommendation ITU-T H.762 (2011), *Lightweight interactive multimedia framework for IPTV services*.

[ITU-T H.763.1] Recommendation ITU-T H.763.1 (2010), *Cascading style sheets for IPTV services*.

[ITU-T H.764] Recommendation ITU-T H.764 (2012), *IPTV services enhanced script language*.

[ITU-T H.770] Recommendation ITU-T H.770 (2015), *Mechanisms for service discovery and selection for IPTV services*.

[ITU-T X.509] Recommendation ITU-T X.509 (2012), *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

[IETF RFC 5246] IETF RFC 5246 (2008), *The Transport Layer Security (TLS) Protocol Version 1.2*.

[IETF RFC 7230] IETF RFC 7230 (2014), *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*.

[W3C Widgets] W3C Recommendation (2012), *Packaged Web Apps (Widgets) – Packaging and XML Configuration (Second Edition)*.

[W3C WidgetsDigSig] W3C Recommendation (2013), *XML Digital Signatures for Widgets*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 content delivery network (CDN) [b-ITU-T F.750]: A network optimized for delivering digital content.

3.1.2 content provider [b-ITU-T Y.1910]: The entity that owns or is licensed to sell content or content assets.

3.1.3 end-user [b-ITU-T Y.1910]: The actual user of the products or services.

NOTE – An end-user consumes the product or service. An end-user can optionally be a subscriber.

3.1.4 key [b-ITU-T X.800]: A sequence of symbols that controls the operations of encipherment and decipherment.

3.1.5 IPTV terminal device [b-ITU-T Y.1901]: A terminal device which has ITF functionality, e.g., a set top box (STB).

3.1.6 metadata [b-ITU-T Y.1901]: Structured, encoded data that describe characteristics of information-bearing entities to aid in the identification, discovery, assessment, and management of the described entities.

3.1.7 multimedia [b-ITU-T J.148]: The combination of multiple forms of media such as audio, video, text, graphics, fax, and telephony in the communication of information.

3.1.8 network provider [b-ITU-T Y.1910]: The organization that maintains and operates the network components required for IPTV functionality.

NOTE 1 – A network provider can optionally also act as service provider.

NOTE 2 – Although considered as two separate entities, the service provider and the network provider can optionally be one organizational entity.

3.1.9 service [b-ITU-T Y.1901]: A set of functionalities enabled by a provider for end-users.

3.1.10 service provider [b-ITU-T M.1400]: A general reference to an operator that provides telecommunication services to customers and other users either on a tariff or contract basis. A service provider may or may not operate a network. A service provider may or may not be a customer of another service provider.

NOTE – In the context of IPTV systems, typically, the service provider acquires or licenses content from content providers and packages this into a service that is consumed by the end-user.

3.1.11 terminal device (TD) [b-ITU-T Y.1901]: An end-user device which typically presents and/or processes the content, such as a personal computer, a computer peripheral, a mobile device, a TV set, a monitor, a VoIP terminal or an audio-visual media player.

3.1.12 user interface (UI) [b-ITU-T F.902]: Software and hardware components through which a user can interact with a system.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 IPTV widget: See packaged IPTV application.

3.2.2 packaged IPTV application: Lightweight application that is intended to be used frequently, such as calendars and news aggregators, with an easily accessible graphical user interface (GUI), often staying on the terminal device display.

3.2.3 widget engine: Software layer that provides features so that widgets can run on a certain user platform (i.e., an IPTV terminal device).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AES	Advanced Encryption Standard
CSS	Cascading Style Sheets
EPG	Electronic Program Guide
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IPTV	Internet Protocol Television
LIME	Lightweight Interactive Multimedia Framework
MAFR	Multimedia Application Framework
NCL	Nested Context Language
RSA	Rivest-Shamir-Adleman
SESL	Services Enhanced Script Language
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security

5 Conventions

None.

6 Introduction

Widget is the definition used for an interactive element of a GUI that displays an information arrangement. The origin of the term "widget" comes from the short for "window gadget", and it was first applied to user interface elements on documents from Project Athena in 1988 [b-Athena]. The term widget is also commonly used to specify an element different from basic GUI components because it provides a single interaction point for the direct manipulation of data in a particular context, but as visual components, widgets can be combined to form an application, or may be used separately, as individual applications. Interactive buttons, sliders, boxes, menus, and other components may be called widgets [b-WidgetsIntro]. However, recently the term is more often used to name a *lightweight application* (e.g., stock market monitor, weather forecast, calculator, news aggregator, etc.) that is usually packaged into a single file to facilitate its provision, transport, and installation.

Web widget, as a "packaged web app" [b-W3C WidgetsReqs], is an interactive single-purpose application for displaying and/or updating local data or data on the Web, packaged in a way to allow a single download and installation on a user's system. However, widgets are used on many environments and with different applicability – they may be found on computers' desktops, mobile devices, web applications, and digital TV platforms as well.

A widget engine is the software layer that supports running and displaying widgets on a GUI, such as the graphical layer of an IPTV terminal device. It is common that such widgets provide relevant information graphically and/or provide easy access to frequently used functions on a system.

6.1 IPTV widgets overview

Packaged IPTV applications, or simply IPTV widgets, are lightweight applications that are used frequently, such as calendars and news aggregators, with an easily accessible GUI that often stay on the display.

IPTV widgets may be classified by their functionality. The classification below is a non-exhaustive list of categories that has been collected from [b-WidgetsIntro], revised and extended to the IPTV domain:

– Accessory widgets: self-contained widgets that do not require support from a content provider or from other applications (e.g., clocks, calculators, offline games);

– Application widgets: widgets that just present a different interface for a regular application already present in the terminal device (e.g., mini player, address book, picture frame);

– Information widgets: widgets that display processed data downloaded from a content provider (e.g., news readers, information tickers, weather forecasters);

– Service widgets: Information widgets that are related to IPTV services (channel-specific electronic program guide (EPG), content recommenders, service provider announcers).

Since an IPTV widget may run on different kinds of terminal devices (like set-top boxes, TV sets, and mobile devices), portability is an important issue and should be addressed based on standardized technologies supported by widget engines in the terminal device. IPTV widgets must be developed using the technologies defined in the ITU-T H.760-series Multimedia Application Framework (MAFR), [ITU-T H.760], such as lightweight interactive multimedia framework (LIME) [ITU-T H.762], CSS [ITU-T H.763.1], SESL [ITU-T H.764], Ginga-NCL [ITU-T H.761], HTML (a new Recommendation is currently under study in ITU-T) and Lua (a new Recommendation is currently under study in ITU-T).

7 IPTV widget service

The widget service takes advantage of functionalities described in [ITU-T H.721] IPTV terminal device basic model, as well as service discovery protocols and services specified in [ITU-T H.770] and in the family of MAFR specifications including [ITU-T H.761] (Ginga-NCL), [ITU-T H.762] (LIME) and [ITU-T H.764] (SESL).

Table 7-1 illustrates the main features of the widget service.

Table 7-1 – Main features of the widget service

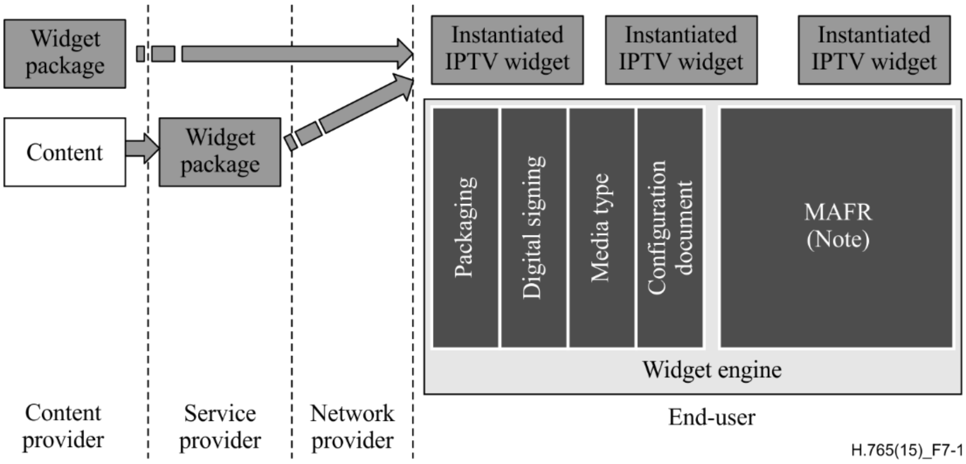
	ITU-T widget service scope	Notes
Widget service discovery	Yes	Refer to clause 7.2, see also [ITU-T H.770]
Interactive widgets	Yes	Based on supported MAFR user agent (Note 1)
Non interactive widgets	Yes	Based on supported MAFR user agent (Note 1)
Media type	A/V, audio, mage, text	Depends on client functions of terminal device (Note 2)

Table 7-1 – Main features of the widget service

	ITU-T widget service scope	Notes
Access local data	Yes	Depends on client functions of terminal device (Note 2) Based on supported MAFR user agent (Note 1)
Access remote web data	Yes	Depends on client functions of terminal device (Note 2) Based on supported MAFR user agent (Note 1)
Widget packaging (for download)	Yes	Refer to clauses 9.1 and 9.2 Based on W3C
GUI	Yes	Depends on client functions of terminal device (Note 2)
NOTE 1 – Currently supported user agents include [ITU-T H.761], [ITU-T H.762] and [ITU-T H.764]. NOTE 2 – Currently supported terminal devices include [ITU-T H.721] and [ITU-T H.722].		

7.1 Widget service end-to-end architecture

Figure 7-1 describes the end-to-end configuration for a widget package execution scenario. The main involved elements are highlighted and described below, as well as the interaction among them.



NOTE –For more details on supported MAFR user agents, see Figure 8-1.

Figure 7-1 – End to end configuration

A widget package is received by the end-user terminal device from a content provider through a service provider and a network provider. The service provider is responsible for organizing a collection of widget packages from different content providers with multiple versions of the widget (e.g., for different scenarios and devices) and different mechanisms for accessing (search, deploy, etc.) the widget packages. All data between the service provider and the end-user terminal device is sent over a network provider. The widget package is optionally associated to an audiovisual content, which may be received from any service provider.

The end-user terminal device is provided with a widget engine, which is the entity responsible for instantiating widgets running on an IPTV terminal device. An IPTV widget engine handles the widget packages, including application unpacking, authenticity and integrity verification, proper configuration of the resources used by the widget, and also the widget's lifecycle control. IPTV widget engines are based on the MAFR technologies as defined by [ITU-T H.760] series. Clause 8 provides further information regarding the IPTV widget engine.

7.2 IPTV widget service discovery

Widget services are discovered using the framework established in [ITU-T H.770]. Widget service metadata can be accessed under the "other services" category defined in [ITU-T H.770]. Metadata semantics specific to the widget service is described in Table 7-2.

Table 7-2 – Metadata semantics specific to the widget service

Element/attribute			Description	Example
Widget Services				
	@DomainName		The unique identifier given to the widget provider	WidgetsAnyCompany.net
	@Version		Version of the offering record	3
	Widget			
		@Id	Identifies an instance of the widget allowing the service provider to provide multiple versions of the widget	231
		Name	Name of the specific widget	CalendarWidget
		Description	Textual description of the widget for potential display. Several textual descriptions are possible, each of them in a specific language	Schedule/calendar service on your IPTV
		Author	Textual description of the widget author (person, or company)	Any company
		License	Usage conditions	(Usage conditions)
		Language	Supported language(s)	en (English)
		Minver	Minimum version of the widget engine required to run the widget	(minimum version to run the widget)
		Maxver	Maximum version of the widget engine required to run the widget	(maximum version to run the widget)
		Rating	Popularity rating of the widget	2 (popularity ranking)
		Downloads	Cumulative number of downloads	1222 (number of downloads)
		Added	Date of addition	2008.6.1 (date of addition)
		Updated	Date of latest update	2009.1.1 (latest update date)
		Picked	Flag set by service provider to promote one or more widgets	1
		Tag	Text related to the widget without any predefined taxonomy	(tags describing widget)
		Locator	Location (e.g. URI or IP addresses) to access the widget	CalendarWidget. AnyCompany.com
	Widget			
		@Id		

Table 7-2 – Metadata semantics specific to the widget service

Element/attribute	Description	Example
Widget Services		
Name		
…		

Widget service metadata can be extended by using a separate namespace (XMLNS) with the definition of other XML elements and attributes. Elements outside the widget service namespace are simply ignored during processing.

The IPTV widget service namespace is `http://www.itu.int/xml-namespace/h.765/widget-service`.

7.3 Protocols for widget service

The IPTV widget service comprises the following steps that demand the standardization of communication protocols and data formats:

– Service discovery: protocols and data formats to announce the IPTV widget service;

– Transaction processing: protocols and data formats to allow the end-user terminal device to list, search, and acquire an IPTV widget. Transaction processing usually involves the exchange of sensitive data;

– Package delivery: protocols and data formats to deliver an IPTV widget package to the end-user terminal device.

Tables 7-3, 7-4 and 7-5 specify the minimum recommended protocols and data formats that an IPTV widget service is required to support.

Table 7-3 – Protocols and data formats for service discovery

Service discovery	Protocol/data format
Service discovery protocols	See [ITU-T H.770]
General service metadata	See [ITU-T H.770]
Specific service metadata	See clause 7.2

Table 7-4 – Protocols and data formats for transaction processing

Transaction processing	Protocol/data format
Transaction processing protocols	HTTPS (SSL3/TLS 1.2) [IETF RFC 5246]
Certificate format	ITU-T X.509
Key exchange / authentication algorithm	RSA (Note)
Key length	2048 bits
Cipher algorithm	AES-256 (Note)
Message authentication	SHA-256 (Note)
Widget metadata	See clause 7.2
NOTE – See more details in [IETF RFC 5246] – TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite.	

Table 7-5 – Protocols and data formats for package delivery

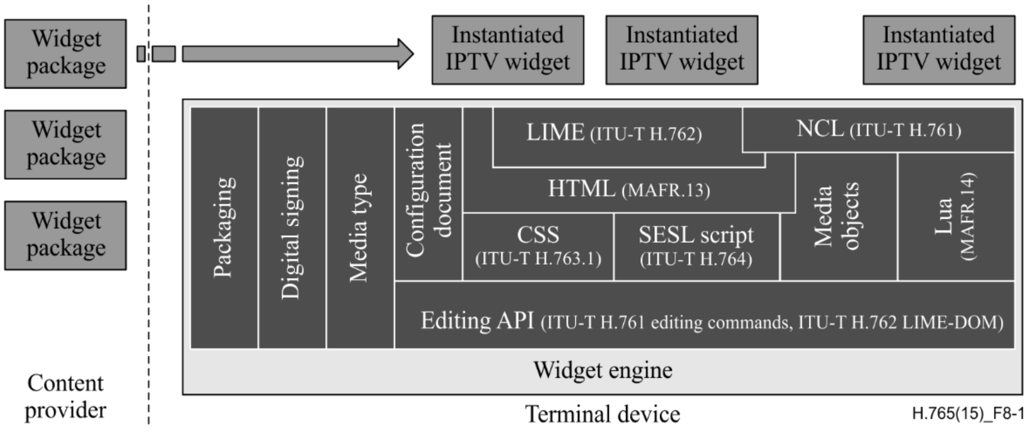
Package delivery	Protocol/data format
Package delivery protocols	HTTP [IETF RFC 7230] HTTPS (SSL3/TLS 1.2) [IETF RFC 5246]
Certificate format	ITU-T X.509
Key exchange / authentication algorithm	RSA (Note)
Key length	2048 bits
Cipher algorithm	AES-256 (Note)
Message authentication	SHA-256 (Note)
Widget package format	See clause 9.1
NOTE – See more details in [IETF RFC 5246] – TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite.	

8 IPTV widget engine

A widget engine is the software layer that provides features so that widgets can run on a certain user platform (i.e., an IPTV terminal device). Also referred as "widget user agents" [b-W3C WidgetsReqs], they make use of a number of technologies that compose an environment to be used by developers to create lightweight applications. To achieve interoperability, widget engines must rely on a set of standards and protocols that enable the rapid development of single-purposed applications that handle multimedia resources (such as images, audio, video, etc.), and are also able to communicate over a network (using hypertext transfer protocol (HTTP), for instance).

An IPTV widget engine is the entity responsible for instantiating widget(s) in the client side (i.e., the IPTV terminal device). Figure 8-1 illustrates the basic concept of a widget engine.

An IPTV widget engine is based on the MAFR technologies defined by [ITU-T H.760] series. The IPTV widget engine handles the widget's instantiation (unpacking and configuring), verifies the widget's authenticity and integrity, allocates the proper resources for the widget's execution (players, API, etc.), and controls the widget's execution lifecycle. As presented in Figure 8-1, an IPTV widget package shall contain NCL ([ITU-T H.761]) and/or LIME ([ITU-T H.762]) application(s), along with the resource files (media, scripts, etc.) within its scope.



NOTE – MAFR.13 and MAFR.14 are currently under study in ITU-T.

Figure 8-1 – Widget and widget engine

In an IPTV scenario, due to its flexibility and openness, presentation concurrency between third-party widgets and service platform provided content is foreseeable. This possibility may lead to discussions that are out of ITU-T Recommendations' scope. The definition of good practices and/or system restrictions to deal with that subject are implementation dependent.

Implementation details dealing with possible resource limitations of the terminal device impacting widget engine performance or behaviour are out of scope.

9 IPTV widgets

It is desirable that IPTV widgets be lightweight, easy to use, fast, accessible, portable, interoperable, customizable, secure, and shareable. These goals can be used to define a set of guidelines for the development of widgets. The following guidelines are independent of the MAFR technology chosen to develop a widget:

– Packaging: a widget must be packaged using a standard format recognizable by IPTV widget engines. Widget developers must keep in mind that this package will be distributed to different devices, locations, languages, and depending of the business model, may be required to be signed.

– Metadata and configuration: widget developers shall have the tools to inform the widget engine the configuration information of a widget, comprising: metadata elements about a widget, such as its title, some form of identification, and versioning information; metadata containing authorship information; a bootstrapping mechanism in order to enable the widget user agents to automatically instantiate a widget; environment configuration parameters.

– Security: the development of widgets involves dealing with security issues, so that the execution of the widget does not harm the terminal device nor compromises data privacy. Using functionalities provided by the supporting widget engine, the developer must guarantee that the widget handles sensitive data confidentially and that it verifies data authenticity and integrity when needed.

8.1 IPTV widget packaging

Widget packaging refers to the technologies related to the encapsulation of all locally necessary resources and metadata required by the widget into a single file so that it can be distributed and deployed. The *de facto* standard for packaging widgets is the packaging format specified in clause 5 of [W3C Widgets].

An IPTV widget package includes the application file (e.g., an NCL or LIME application document), which comprises a configuration file and many media files (images, videos, etc.) compressed into a single file, using the compression algorithm defined in clause 5 of [W3C Widgets]. The package file extension is recommended to be ".iwp", which stands for [ITU-T IPTV] widget package. In this case, the media type must be "application/x-itu-iptv-widget".

Any IPTV widget package must contain one configuration file which is an XML document containing metadata information and configuration parameters to be interpreted by the widget engine. The configuration file must be placed in the root level of the package, and it must be named config.xml. The next clause describes the information provided by an IPTV widget configuration file.

9.2 IPTV widget metadata/configuration

IPTV widget metadata/configuration is related to how the information regarding a widget is stored inside the package, and how this data is made accessible. Information necessary for IPTV widget configuration is provided in the form of an XML document containing relevant metadata.

Common metadata included in the widget configuration document is shown in Table 9-1. In order to support proprietary extensions of widget metadata, additional elements or attributes to those shown in the table can be included using a separate namespace. If the widget engine supports the corresponding element or attribute, then it is required to process accordingly, otherwise the widget engine is required to ignore.

The widget namespace is: `http://www.itu.int/xml-namespace/h.765/widget`.

Table 9-1 – Common metadata included in the widget configuration document

Element / @Attribute		Description	Value example (explanation)	Optionality
Widget				M
	@id	URI that uniquely identifies the widget. Recommended to be an accessible URI for reference purposes, update checking, notifications, etc.	http://example.org/exampleWidget	O
	@width	Preferred width for the widget exhibition area	200	O
	@height	Preferred height for the widget exhibition area	320	O
	@viewmodes	List of attributes denoting the preferred way of displaying the widget to the end-user. The list contains one or more values, separated by space characters, ordered by preference where the first value denotes the preferred way. Possible values relevant to IPTV terminal devices: windowed; floating; fullscreen; minimized	Windowed floating	O
	@version	A string that identifies the version of the widget	2.1	O
	name	A human readable string used, for instance, to identify the widget in a menu. Several	Example widget	O

Table 9-1 – Common metadata included in the widget configuration document

Element / @Attribute			Description	Value example (explanation)	Optionality
			names are possible, each of them in a specific language. Use attribute xml:lang to specify the name language		
	description		A human readable string for a textual description of the widget. Several textual descriptions are possible, each of them in a specific language. Use attribute xml:lang to specify the description language	Example service on your IPTV	O
	author		Textual description of the widget author (person, or company)	Any Company	O
		@href	Author's website URI	http://www.anycompany.com	O
		@email	Author's Email address	widgets@anycompany.com	O
	license		String containing the copyright information for the widget	Copyright 2010 by Any Company This program is free software; you can redistribute…	O
	language		Supported language(s)	en (English)	O
	icon@Src		Icon image file for the widget	/media/icon.png	O
	content				M
		@src	Path to the main application source file	/main.ncl (relative path from the root of the package file system)	M
		@type	Type of the main application source file	application/x-ginga-ncl	O
	tag		Text related to the widget without any predefined taxonomy	(tags describing widget)	O
	feature		URI to be used by the user agent to request a binding to a runtime component	Runtime components needed by the widget (e.g., a video decoder)	O
		@name	URI attribute	http://www.anycompany.com	O
		@required	Boolean value to specify if the component is required to function properly	"true": Widget needs to support the feature to function "false": Widget can function without supporting the feature The default value when the attribute is omitted is "false"	O
	param		Defines a parameter for the corresponding feature using name-value pair	param is a child of the feature	O
		@name	String to denote the name of the parameter	name= "Update frequency" Useful widget information (e.g., stock or weather widget)	O

Table 9-1 – Common metadata included in the widget configuration document

Element / @Attribute	Description	Value example (explanation)	Optionality
@value	Value of the parameter	value= "Once per hour"	O
preference	Persistent storage for preferences using a name-value pair		O
@name	String to denote the name of the preference	name= "Home menu"	O
@value	Value of the preference	value= "My preferred skin"	O
@readonly	Boolean value to allow, disallow overwriting	"true": cannot be overwritten at runtime "false": can be overwritten The default value when the attribute is omitted is "false"	O

The extension of the configuration document format can be done by using a separate namespace (XMLNS) with the definition of other XML elements and attributes. Elements outside the widget namespace are simply ignored during processing.

The following is an example illustrating how to include proprietary metadata using an external namespace. In this example, in addition to the standard namespace an external namespace ("`http://somecompany.com/`") enables the use of proprietary metadata. Proprietary metadata (i.e., not included in Table 9-1 of clause 9.2) can be used as shown in the fragment below. The attribute "`src`" is defined in the standardized namespace, while the proprietary attribute "role" is defined in the proprietary namespace.

```
<widget xmlns="http://www.itu.int/xml-namespace/h.765/widget"
   xmlns:ex="http://somecompany.com/">

      <icon src="big.png" ex:role="big"/>
      <ex:datasource>{'a':'b','c':'d'}</ex:datasource>

      <content src="widget.bml"/>
</widget>
```

9.3 IPTV widget security

IPTV widget security mechanisms are required to address two relevant aspects:

– The protection of the package that contains a widget for a secure installation in terminal devices;

– The protection of sensitive data that widgets exchange with their associated servers.

9.3.1 Package protection

Packaging function is required to support package protection mechanisms, which ensure the following security aspects:

– Authentication: the package has been created by the widget's identified content provider;

– Integrity: the package has not been modified from the content provider up to the end-user terminal device;

– Non-repudiation: the content provider cannot at a later time deny having created the package.

These mechanisms are required to be included during the packaging of a widget under the form of digital signatures. IPTV widget digital signatures are required to be compliant with W3C XML digital signatures for widgets [W3C WidgetsDigSig]. Table 9-2 specifies the minimum recommended set of algorithms and formats for IPTV widget digital signing.

Table 9-2 – Specifications for package protection: digital signing

Package protection	Specification
Certificate format	[ITU-T X.509]
Signature algorithm	RSA-SHA256 (Note)
Key length	4096 bits
Digest method	SHA-256 (Note)
NOTE – See more details in [W3C WidgetsDigSig].	

9.3.2 Data exchange protection

IPTV widgets might exchange sensitive data with their associated servers. Any data exchange is implemented in a widget by using the communication APIs provided by the underlying MAFR technology of a widget engine. Therefore, widget data exchange protection is required to be supported by the chosen MAFR technology. Table 9-3 specifies the minimum recommended set of protocols, algorithms and formats for IPTV widget secure communications.

Table 9-3 – Specifications for data exchange protection: secure communications

Data exchange protection	Specification
Secure protocol	HTTPS (SSL3/TLS 1.2) [IETF RFC 5246]
Certificate format	[ITU-T X.509]
Key exchange / authentication algorithm	RSA (Note)
Key length	2048 bits
Cipher algorithm	AES-256 (Note)
Message authentication	SHA-256 (Note)
NOTE – See more details in [IETF RFC 5246] – TLS_RSA_WITH_AES_256_CBC_SHA256 cipher suite.	

Appendix I

ITU-T IPTV widgets versus W3C widgets

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

This appendix is intended to provide a comparison between the definitions for widgets from W3C and ITU-T IPTV.

Table I.1 – Comparison between W3C's and ITU-T IPTV's widget definitions

	W3C	**This ITU-T IPTV scope**
What is a widget?	Application	Lightweight application
Widget service	Not defined	Yes, see clause 7
Interactivity	Yes	Yes, but also widgets without interactivity
Single purpose	Yes	Not necessarily, so leave it out of the definition
Display data	Yes	Yes, but could be audible data, therefore use "render" data
Access local data	Yes	Yes
Access remote web data	Yes	Yes
Packaged for download	Yes	Yes
Easily accessible	No mention	Yes
GUI	No mention	Yes

Appendix II

IPTV widget examples

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

II.1 ITU-T H.761 widget example

A simple widget example is presented in this clause. The configuration file defines a start NCL application that exhibits news – the content is reloaded when the user presses the "BLUE" key and the application exits when the "RED" key is pressed. The configuration file defines a specific value for the "s*ervice.language*" variable, which shall be used by the execution environment – all configuration parameters set by the configuration file must be also referenced by the application (the NCL application must include the same media objects for the associated parameters). In the example below, the widget (e.g., *news-widget.iwp*) contains the following directory tree:

- main.ncl
- config.xml
- css/
- css/style.css
- media/
- scripts/
- scripts/counter.lua

config.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns       = "http://www.itu.int/xml-namespace/h.765/widget"
        id          = "http://example.org/exampleWidget"
        version     = "3.0"
        height      = "300"
        width       = "200">
  <name>
    Example Widget
  </name>
  <description>
    A sample widget.
  </description>
  <author href  = " http://www.itu.int/ITU-T/gsi/iptv/"
          email = "moreno@ice.ufjf.br">Marcelo Moreno</author>
  <icon src="icons/icon.png"/>
  <content src="main.ncl" type="application/x-ginga-ncl"/>
  <license>
        Copyright.2010. International Telecommunication Union
  </license>
</widget>
```

main.ncl file

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ncl id="test_widget" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
   <head>
      <!-- widget size and position relative to those defined by config.xml -->
      <regionBase>
         <region id="main_region" left="10" top="10" width="300" height="400"/>
      </regionBase>
      <!-- Stylesheet -->
      <descriptorBase>
```

```
         <descriptor id="main_content" region="main_region"
                     style="css/styles.css"/>
      </descriptorBase>
      <connectorBase>
         <causalConnector id="onKeySelectionStopStartSet">
            <connectorParam name="keyCode"/>
            <connectorParam name="var"/>
            <simpleCondition role="onSelection" key="$keyCode" />
            <compoundAction operator="seq">
               <simpleAction role="stop" max="unbounded" qualifier="par"/>
               <simpleAction role="start" max="unbounded" qualifier="par"/>
               <simpleAction role="set" value="$var" max="unbounded"
                             qualifier="par"/>
            </compoundAction>
         </causalConnector>
         <causalConnector id="onKeySelectionStopN">
            <connectorParam name="keyCode"/>
            <connectorParam name="var"/>
            <simpleCondition role="onSelection" key="$keyCode" />
            <simpleAction role="stop" max="unbounded"/>
         </causalConnector>
      </connectorBase>
      <!-- Rules used for internationalization of the content -->
      <ruleBase>
         <rule id="r_en" var="service.language" comparator="eq" value="eng"/>
         <rule id="r_pt" var="service.language" comparator="eq" value="por"/>
      </ruleBase>
   </head>
   <body>
      <port id="main_port" component="html_content"/>
      <!-- 'settings' node, repeats the parameters from config.xml -->
      <media type="application/x-ncl-settings" id="settings">
         <property name="service.language"/>
      </media>
      <!-- Lua script counting number of reloads -->
      <media id="lua_counter"  src="scripts/counter.lua">
         <property name="inc"/>
      </media>
      <!-- Content based on the presentation environment language -->
      <switch id="html_content">
         <defaultComponent component="html_en" />
         <bindRule rule="r_pt" constituent="html_pt" />
         <media id="html_pt" src="http://www.ginga.org.br/noticias.php"
                type="text/html" descriptor="main_content"/>
         <media id="html_en" src="http://www.ginga.org.br/news.php"
                type="text/html" descriptor="main_content"/>
      </switch>
      <!-- Ends the widget when users presses the red button -->
      <link xconnector="onKeySelectionStopN">
         <bind component="html_content" role="onSelection">
            <bindParam name="keyCode" value="RED"/>
         </bind>
         <bind role="stop" component="html_content"/>
      </link>
      <!-- The blue key reloads the news pages and increases the counter -->
      <link xconnector="onKeySelectionStopStartSet">
         <bind role="onSelection" component="html_content">
            <bindParam name="keyCode" value="BLUE"/>
         </bind>
         <bind role="stop" component="html_content"/>
         <bind role="start" component="html_content"/>
         <bind role="set" component="lua_counter" interface="inc">
            <bindParam name="var" value="1"/>
         </bind>
      </link>
   </body>
</ncl>
```

counter.lua file

```
counter = 0
local counterEvt = {
    class = 'ncl',
    type  = 'attribution',
    property = 'counter',
}

function handler (evt)
    if evt.class ~= 'ncl' then return end
    if evt.type ~= 'attribution' then return end
    if evt.property ~= 'inc' then return end

    counter = counter + evt.value

    event.post {
        class    = 'ncl',
        type     = 'attribution',
        property = 'inc',
        action   = 'stop'
    }

    counterEvt.value = counter
    counterEvt.action = 'start'; event.post(counterEvt)
    counterEvt.action = 'stop';  event.post(counterEvt)
end

event.register(handler)
```

II.2 ITU-T H.762 widget example

An example of a simple ITU-T H.762 widget is presented in this clause. The configuration file `config.xml` follows the conventions defined in clause 7.2 including necessary elements and attributes. In this example the files included in the widget package are listed below. Note that this particular widget example does not require any external script file.

- config.xml
- startup.bml
- object-element-test1.lime
- media/
- media/lime-large.jpg
- media/lime-small.jpg
- media/ITU-icon.jpg
- scripts/

The content of the files in this example with the exception of the media files are presented below.

config.xml file

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<widget xmlns = "http://www.itu.int/xml-namespace/h.765/widget" id="123"
width="200" height="200" version="1">
  <Name>Widget sample in LIME</Name>
  <Description>Simple LIME Widget sample,
               with two JPEG files.
  </Description>
  <Author href="http://www.itu.int/ITU-T/gsi/iptv/"
          email="john.doe@ties.itu.int">
      John Doe
  </Author>
  <License>Copyright.2010. International Telecommunication
           Union
  </License>
  <Language>English, Japanese</Language>
  <Icon src="icon/ITU-logo.jpg" />
  <Content src="startup.bml" type="application/x-lime" />
  <Tag> images, LIME <Tag/>
</Widget>
```

startup.bml file

```
<?xml version="1.0" encoding="EUC-JP" ?>
<!DOCTYPE bml PUBLIC "-//ARIB STD-B24:1999//DTD BML Document for IPTV//JA"
"http://www.arib.or.jp/B24/DTD/bml_x_x_iptv.dtd">
<?bml bml-version="100.0" ?>
<bml>
<head>
<title>startup -- bml version</title>
<script><![CDATA[

function onload() {
browser.launchDocument("lime/object-element-test1.lime", "cut");
}

]]></script>

</head>
<body id="body" onload="onload();">

</body>
</bml>
```

object-element-test1.lime file

```
<?bml bml-version="100.0" ?>

<bml>

<head>
<title>Object element test 1:JPEG</title>
</head>

<body style="background-color-index:7">

<div style="left:0px;top:0px;width:960px;height:500px">
<object data="../images/lime-large.JPG" type="image/jpeg"
style="width:369px;height:252px;left:300px;top:150px;"/>
<object data="../images/ITU-Logo.JPG" type="image/jpeg"
style="width:189px;height:77px;left:20px;top:10px;"/>
</div>
</body>
</bml>
```

Bibliography

[b-ITU-T F.750] Recommendation ITU-T F.750 (2005), *Metadata framework*.

[b-ITU-T F.902] Recommendation ITU-T F.902 (1995), *Interactive services design guidelines*.

[b-ITU-T J.148] Recommendation ITU-T J.148 (2003), *Requirements for an objective perceptual multimedia quality model*.

[b-ITU-T M.1400] Recommendation ITU-T M.1400 (2015), *Designations for interconnections among operators' networks*.

[b-ITU-T X.800] Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

[b-ITU-T Y.1901] Recommendation ITU-T Y.1901 (2009), *Requirements for the support of IPTV services*.

[b-ITU-T Y.1910] Recommendation ITU-T Y.1910 (2008), *IPTF functional architecture*.

[b-Athena] George A. Champine, Daniel E. Geer, and William N. Ruh (1990), '*Project Athena as a Distributed Computer System*,' IEEE Computer 23(9), pp. 40-50 (September).

[b-W3C WidgetsReqs] W3C Working Draft (2009) – *Widgets 1.0 Requirements*. <http://www.w3.org/TR/2009/WD-widgets-reqs-20090430>.

[b-WidgetsIntro] C Kaar (2007), *An introduction to Widgets with particular emphasis on Mobile Widgets Computing*, October.

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
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2015