ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T J.1306 (07/2023)

SERIES J: Cable networks and transmission of television, sound programme and other multimedia signals

Cloud-based converged media services for IP and broadcast cable television

Specification of microservices architecture for audiovisual media in the converged media cloud

ITU-T J-SERIES RECOMMENDATIONS

Cable networks and transmission of television, sound programme and other multimedia signals

GENERAL RECOMMENDATIONS	J.1-J.9
GENERAL SPECIFICATIONS FOR ANALOGUE SOUND-PROGRAMME TRANSMISSION	J.10-J.19
PERFORMANCE CHARACTERISTICS OF ANALOGUE SOUND-PROGRAMME CIRCUITS	J.20-J.29
EQUIPMENT AND LINES USED FOR ANALOGUE SOUND-PROGRAMME CIRCUITS	J.30-J.39
DIGITAL ENCODERS FOR ANALOGUE SOUND-PROGRAMME SIGNALS – PART 1	J.40-J.49
DIGITAL TRANSMISSION OF SOUND-PROGRAMME SIGNALS	J.50-J.59
CIRCUITS FOR ANALOGUE TELEVISION TRANSMISSION	J.60-J.69
ANALOGUE TELEVISION TRANSMISSION OVER METALLIC LINES AND INTERCONNECTION WITH RADIO-RELAY LINKS	J.70-J.79
DIGITAL TRANSMISSION OF TELEVISION SIGNALS	J.80-J.89
ANCILLARY DIGITAL SERVICES FOR TELEVISION TRANSMISSION	J.90-J.99
OPERATIONAL REQUIREMENTS AND METHODS FOR TELEVISION TRANSMISSION	J.100-J.109
INTERACTIVE SYSTEMS FOR DIGITAL TELEVISION DISTRIBUTION (DOCSIS FIRST AND SECOND GENERATIONS)	J.110-J.129
TRANSPORT OF MPEG-2 SIGNALS ON PACKETIZED NETWORKS	J.130-J.139
MEASUREMENT OF THE QUALITY OF SERVICE – PART 1	J.140-J.149
DIGITAL TELEVISION DISTRIBUTION THROUGH LOCAL SUBSCRIBER NETWORKS	J.150-J.159
IPCABLECOM (MGCP-BASED) – PART 1	J.160-J.179
DIGITAL TRANSMISSION OF TELEVISION SIGNALS – PART 1	J.180-J.189
CABLE MODEMS AND HOME NETWORKING	J.190-J.199
APPLICATION FOR INTERACTIVE DIGITAL TELEVISION – PART 1	J.200-J.209
INTERACTIVE SYSTEMS FOR DIGITAL TELEVISION DISTRIBUTION (DOCSIS THIRD TO FIFTH GENERATIONS)	J.210-J.229
MULTI-DEVICE SYSTEMS FOR CABLE TELEVISION	J.230-J.239
MEASUREMENT OF THE QUALITY OF SERVICE – PART 2	J.240-J.249
DIGITAL TELEVISION DISTRIBUTION THROUGH LOCAL SUBSCRIBER NETWORKS	J.250-J.259
IPCABLECOM (MGCP-BASED) – PART 2	J.260-J.279
DIGITAL TRANSMISSION OF TELEVISION SIGNALS – PART 2	J.280-J.289
CABLE SET-TOP BOX	J.290-J.299
APPLICATION FOR INTERACTIVE DIGITAL TELEVISION – PART 2	J.300-J.309
MEASUREMENT OF THE QUALITY OF SERVICE – PART 3	J.340-J.349
IPCABLECOM2 (SIP-BASED) – PART 1	J.360-J.379
DIGITAL TRANSMISSION OF TELEVISION SIGNALS – PART 3	J.380-J.389
MEASUREMENT OF THE QUALITY OF SERVICE – PART 4	J.440-J.449
IPCABLECOM2 (SIP-BASED) – PART 2	J.460-J.479
DIGITAL TRANSMISSION OF TELEVISION SIGNALS – PART 4	J.480-J.489
TRANSPORT OF LARGE SCREEN DIGITAL IMAGERY	J.600-J.699
SECONDARY DISTRIBUTION OF IPTV SERVICES	J.700-J.799
MULTIMEDIA OVER IP IN CABLE	J.800-J.899
TRANSMISSION OF 3-D TV SERVICES	J.900-J.999
CONDITIONAL ACCESS AND PROTECTION	J.1000-J.1099
SWITCHED DIGITAL VIDEO OVER CABLE NETWORKS	J.1100-J.1119
SMART TV OPERATING SYSTEM	J.1200-J.1209
IP VIDEO BROADCAST	J.1210-J.1219
CLOUD-BASED CONVERGED MEDIA SERVICES FOR IP AND BROADCAST CABLE TELEVISION	**J.1300-J.1309**
TELEVISION TRANSPORT NETWORK AND SYSTEM DEPLOYMENT IN DEVELOPING COUNTRIES	J.1400-J.1409
ARTIFICIAL INTELLIGENCE (AI) ASSISTED CABLE NETWORKS	J.1600-J.1649

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

Recommendation ITU-T J.1306

Specification of microservices architecture for audiovisual media in the converged media cloud

Summary

Recommendation ITU-T J.1306 specifies the architecture and related components of audiovisual media based on microservice technologies. Recommendation ITU-T J.1306 fulfils the requirements in Recommendation ITU-T J.1305. This Recommendation applies to the design, development, construction, operation and maintenance of audiovisual media systems based on microservices. Recommendation ITU-T J.1306 is based on the characteristics of microservice technology and audiovisual media business, combines the requirements of technology and business, and devises an audiovisual media microservice architecture (MMA) that meets the needs of rapid iteration and diversified services of the audiovisual media business. MMA follows the layered architectural methodology specified in Recommendation ITU-T J.1302. Mainly from the perspective of cloud platform, Recommendation ITU-T J.1302 stipulates the system architecture of cloud-based converged media services (CBCMSs) to support Internet protocol and broadcast cable television services. From the perspective of microservice, Recommendation ITU-T J.1306 specifies the microservices architecture of integrated media based on container, virtual machine, cloud and other infrastructures to support the audiovisual media business carried out by microservices on a variety of infrastructures. From the perspective of microservice governance, Recommendation ITU-T J.1306 realizes the compatibility of various current mainstream microservice frameworks and stipulates the management capabilities of distributed systems. From the media business perspective, it specifies the microservice components that support production, broadcasting, transmission, distribution, interaction and other audiovisual media business. From the perspective of application integration, it stipulates the service orchestration capabilities and the means of application assembly.

History [*]

Edition	Recommendation	Approval	Study Group	Unique ID
1.0	ITU-T J.1306	2023-07-14	9	11.1002/1000/15584

Keywords

CBCMS, cloud-based converged media services, converged media, microservices architecture, specification.

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

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

Table of Contents

Page

Introduction

This Recommendation is part 2 of a multi-part deliverable covering microservices architecture for audiovisual media in converged media cloud, identified as follows:

Part 1: [ITU-T J.1305] – Requirements

Part 2: ITU-T J.1306 – Specification

Recommendation ITU-T J.1306

Specification of microservices architecture for audiovisual media in the converged media cloud

1 Scope

This Recommendation specifies the architecture and related components of audiovisual media based on microservice technologies. This Recommendation fulfils the requirements of [ITU-T J.1305].

This Recommendation applies to the design, development, construction, operation and maintenance of audiovisual media systems based on microservices.

This Recommendation bears the characteristics of microservice technology and audiovisual media business, combines the requirements of technology and business, and comes up with an audiovisual media microservice architecture (MMA) that meets the needs of rapid iteration and diversified services of the audiovisual media business. An MMA follows the layered architectural methodology as specified in [ITU-T J.1302].

Mainly from the perspective of cloud platform, [ITU-T J.1302] stipulates the system architecture of cloud-based converged media services (CBCMSs) to support Internet protocol (IP) and broadcast cable television services. From the perspective of microservice, this Recommendation specifies the microservices architecture of integrated media based on container, virtual machine, cloud and other infrastructures to support the audiovisual media business carried out by microservices on a variety of infrastructures.

From the perspective of microservice governance, this Recommendation realizes the compatibility of various current mainstream microservice frameworks and stipulates the management capabilities of distributed systems. From the media business perspective, it specifies the microservice components that support the production, broadcasting, transmission, distribution, interaction and other audiovisual media business. From the perspective of application integration, it stipulates the service orchestration capabilities and the means of application assembly.

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 J.1302] Recommendation ITU-T J.1302 (2022), *Specification of cloud-based converged media service to support Internet protocol and broadcast cable television – System architecture.*

[ITU-T J.1305] Recommendation ITU-T J.1305 (2023), *Requirements of microservice architecture for audiovisual media in the converged media cloud.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 API gateway [ITU-T J.1302]: In microservices architecture, applications and services are composed of smaller, exchangeable components, and these components need a way to find and communicate with one another. This is where API gateways come in. An API gateway sits between clients and services. It acts as a reverse proxy, routing requests from clients to services. It may also perform various cross-cutting tasks such as authentication, transport layer security (TLS) termination, and rate limiting.

3.1.2 audiovisual media [b-ISO/TR 11219]: Documents in which sound and/or pictures are prominent, and which require the use of special equipment to be viewed and/or heard.

NOTE – This includes audio documents, pictures and films on different physical carriers.

3.1.3 cloud-based converged media service (CBCMS) [b-ITU-T J.1301]: A service intended for deployment by cable television operators and to support the requirements of rapid service innovation and deployment, which enables media services to be developed by separate vendors according to standard application programming interfaces, where appropriate.

3.1.4 container [b-ITU-T Y.3535]: A set of software to provide isolation, resource control and portability for virtualization processing of an application.

3.1.5 converged media [ITU-T J.1305]: New type of media that effectively combines the information technology of radio, television, newspapers and periodicals, network audiovisual and other aspects, with the help of diversified communication channels and forms.

3.1.6 data warehouse [b-ISO 19297-1]: Special kind of database system built upon existing operational databases that plays a key role in building a decision support system for an organization or an enterprise.

3.1.7 digital rights management [b-ITU-T X.1193]: A synonym for service and content protection or content protection, depending upon the context of use.

3.1.8 edge cloud [b-ITU-T J.1303]: The cloud that is deployed close to users' locations and has a limited set of services compared to the central cloud.

3.1.9 media microservice architecture (MMA) [ITU-T J.1305]: A system construction method oriented to audiovisual media based on microservice technologies. Its core idea is to adopt miniaturized and distributed methodologies to provide services to users in system development, deployment and maintenance for complex application scenarios related to audiovisual media, to avoid the problem that the system is difficult to maintain and upgrade with the increasing number of business scenarios.

3.1.10 microservice [b-ITU-T J.1301]: An architectural and organizational approach to software development where software is composed of small independent microservices that communicate over well-defined application programming interfaces. Microservice architecture makes applications easier to scale and faster to develop, enabling innovation and accelerating time-to-market for new features.

3.1.11 microservices architecture [b-ISO/IEC TS 23167]: Design approach that divides an application into a set of microservices.

3.1.12 process engine; workflow engine [ITU-T J.1305]: Software service or "engine" that provides the run time execution environment for a process instance.

NOTE – Adopted from the definition of "workflow engine" in [b-ISO 12651-2].

3.1.13 rule engine [ITU-T J.1305]: A management service for parsing, invoking and executing rules that use predefined semantic modules to write business decisions, accept data input, interpret business rules and make business decisions.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 API lifecycle management: Set of functions required to manage the instantiation, maintenance and termination of an application programming interface instance.

3.2.2 cloud database: A database that is optimized or deployed to a cloud-computing environment.

3.2.3 container orchestration: Procedure on container deployment and organization that provides the ability to schedule and manage container clusters, including container automated deployment, management, elastic scaling and container network management.

3.2.4 continuous integration and continuous deployment: Software engineering practice that helps team members integrate and publish their work results frequently, with continuous integration that automatically conducts error verification and shortens the system development lifecycle, and continuous deployment that automatically releases the verified code and builds system deployed into different targeting environments.

3.2.5 data lake: A large data warehouse that centrally stores all kinds of structured and unstructured data.

3.2.6 development and operations (DevOps): A group of processes, methods and systems collectively used to promote communication, collaboration and integration among application development and system, technology operation and maintenance as a single whole engineering process.

3.2.7 fitness function: An objective function used to calculate the gap between potential solutions and established goals in evolutionary computation paradigm.

3.2.8 grayscale release capability: A way of releasing a smooth transition between black and white.

3.2.9 lakehouse: An integrated system that connects and integrates the data warehouse and data lake, whose coexistence can ensure data flow, reduce repeated construction and bring more benefits to enterprises.

3.2.10 low code development platform: Development platform that can quickly build applications via a graphical user interface, using drag and drop visual method and model-driven logic, without coding or through a small number of codes.

3.2.11 open application model: A paradigm for describing and standardizing applications. During the maintenance lifecycle of applications, it connects application developers, application operation and maintenance personnel, and infrastructure operation and maintenance personnel in a standardized way by providing a uniform method of communication, so as to make the development, delivery and operation, and maintenance of cloud native applications more concise, efficient and controllable.

3.2.12 service mesh: A group of agent components (called sidecars as proxy for each service) and task management components (categorized as control plane and data plane) that deal with a large number of inter-process network communication among microservices.

3.2.13 webhook: A reverse application programming interface (API) invocation concept mainly for asynchronous web construction which provides API rules to send hypertext transfer protocol requests to applications with hooks (often simply a uniform resource locator or API).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI	Artificial Intelligence
API	Application Programming Interface
BFF	Backend For Frontend
CBCMS	Cloud-Based Converged Media Service
CPU	Central Processing Unit
DevOps	Development and Operations
HTTP	Hypertext Transfer Protocol
ID	Identifier
IP	Internet Protocol
MMA	Media Microservice Architecture
SPI	Service Provider Interface
TLS	Transport Layer Security
UI	User Interface
XML	extensible Markup Language

5 Conventions

In this Recommendation:

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

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

The phrase "**is prohibited**" indicates a requirement that must be strictly followed and from which no deviation is permitted if conformance to 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 optionally be enabled by the network operator or service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

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

6 Overall architecture

The overall architecture of MMA (see Figure 1) includes five layers for: infrastructure adaptation; microservice governance capability; media business service; platform service; and application integration. The bottom two layers, for infrastructure adaptation and microservice governance capability, are mainly responsible for basic resource maintenance and invocation, especially the governance of microservices. The top three layers, for media business service, platform service and application integration, are mainly responsible for providing basic business microservices on the media business service layer and forming platform services and application assembly capability through weaving basic business microservices together.

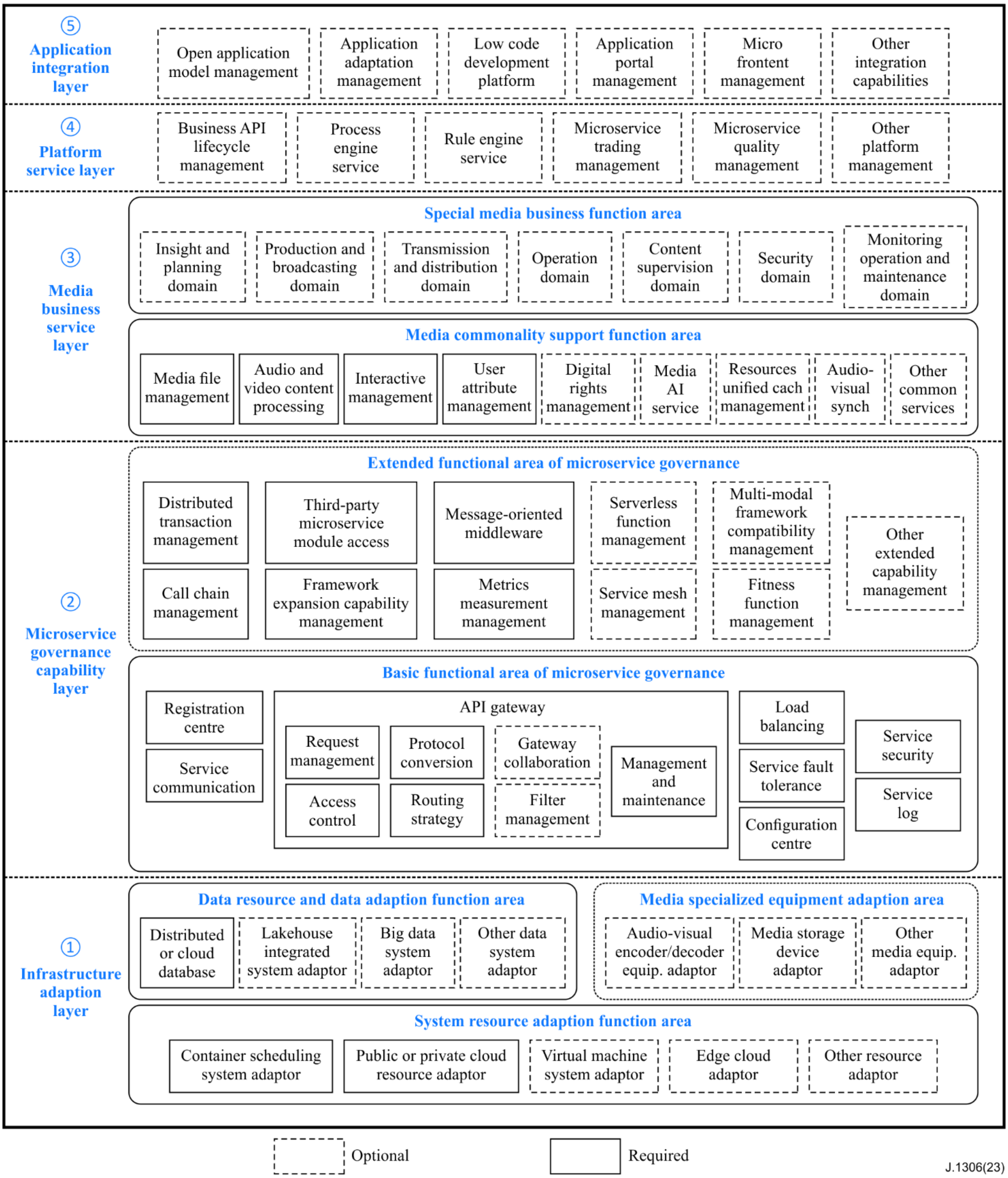
⑤ Application integration layer
Open application model management
Application adaptation management
Low code development platform
Application portal management
Micro frontent management
Other integration capabilities
④ Platform service layer
Business API lifecycle management
Process engine service
Rule engine service
Microservice trading management
Microservice quality management
Other platform management
③ Media business service layer
Special media business function area
Insight and planning domain
Production and broadcasting domain
Transmission and distribution domain
Operation domain
Content supervision domain
Security domain
Monitoring operation and maintenance domain
Media commonality support function area
Media file management
Audio and video content processing
Interactive management
User attribute management
Digital rights management
Media AI service
Resources unified cach management
Audio-visual synch
Other common services
② Microservice governance capability layer
Extended functional area of microservice governance
Distributed transaction management
Third-party microservice module access
Message-oriented middleware
Serverless function management
Multi-modal framework compatibility management
Other extended capability management
Call chain management
Framework expansion capability management
Metrics measurement management
Service mesh management
Fitness function management
Basic functional area of microservice governance
Registration centre
Service communication
API gateway
Request management
Protocol conversion
Gateway collaboration
Access control
Routing strategy
Filter management
Management and maintenance
Load balancing
Service fault tolerance
Configuration centre
Service security
Service log
① Infrastructure adaption layer
Data resource and data adaption function area
Distributed or cloud database
Lakehouse integrated system adaptor
Big data system adaptor
Other data system adaptor
Media specialized equipment adaption area
Audio-visual encoder/decoder equip. adaptor
Media storage device adaptor
Other media equip. adaptor
System resource adaption function area
Container scheduling system adaptor
Public or private cloud resource adaptor
Virtual machine system adaptor
Edge cloud adaptor
Other resource adaptor
Optional
Required
J.1306(23)

Figure 1 – Overall architecture of audiovisual media microservice

7 Infrastructure adaption layer

This layer is required to adapt to fundamental resources for system operation and data, so that the layers above have an operational environment.

7.1 System resource adaption function area

The system resource adaption function area calls the underlying infrastructure (computing, storage, network, etc.), and should provide adaption to the following basic resources:

- container scheduling system adaptor: provides the ability to adapt to the container scheduling system, including the ability to adapt to central processing unit (CPU) or graphic processing unit resources;
- public or private cloud resource adaptor: provides an adaptor to a public or private cloud; can optionally provide cross-cloud, multi-cloud and hybrid cloud resource docking capabilities, supports the interface of different clouds in the form of plug-ins, and manages different clouds on a single platform.

According to requirements, more abundant basic resources can be connected through the following adaptations:

- virtual machine system adaptor: provides the ability to adapt to the virtual machine system;
- edge cloud adaptor: cooperates with the central cloud resource, acts as secondary cloud resource(s) to be near the end users.

7.2 Data resource and data adaption function area

Data resource is required, and data adaption is recommended to provide the following data support capabilities:

- distributed or cloud database: is required to provide the ability of a distributed or cloud database and generate a global unique identifier;
- lakehouse integrated system adaptor: provides the ability to adapt to data warehouse or data lake; it is recommended that the lakehouse system chosen has real-time data processing and online or offline integration capability;
- big data system adaptor: provides the ability to adapt to the big data system that processes data streams.

7.3 Media specialized equipment adaption area

The media industry has a large and varied number of pieces of professional equipment, which are recommended for adaption into MMA (equipment not based on IP first needs conversion). Typical devices follow:

- audiovisual encoder and decoder equipment adaptor: provides audio and video encoding and decoding acceleration processing capability;
- media storage device adaptor: can store and retrieve large media files;
- other media equipment adaptor: e.g., specialized media network transmission equipment that provides accelerated processing or mass distribution capabilities suitable for large media flows and more.

8 Microservice governance capability layer

8.1 Basic functional area of microservice governance

8.1.1 Registration centre

The registration centre should provide high availability, covering registration and discovery capabilities. It is required to obtain the target address of each service instance in the distributed environment and regularly update its current state, so as to enable each microservice instance to initiate remote calls using the service discovery mechanism:

- service registration, service renewal and service offline management capabilities;
- mechanism for service discovery;
- capabilities to maintain the service registry or library;

- support lossless publishing, microservice instance can only be registered after it is confirmed that it can provide services;
- recommended to support container-choreographed service registration and discovery capabilities.

8.1.2 Service communication

The communication between microservices is required to support the following requirements:

- synchronous communication capability: external API calls are recommended for implementation in representational state transfer mode, and internal microservices (especially east-west microservice traffic) communication is carried out in remote procedure call mode;
- protocol extension capability (such as through service provider interface (SPI) plug-ins) to manage the new protocol;
- recommended to support asynchronous communication capability (message system or task management, such as asynchronous message protocol Java message service/advanced message queuing protocol and real-time message service protocol extensible messaging and presence protocol/message queuing telemetry transport);
- recommended to support the protocol between the terminal and the media gateway, such as the signalling control of voice over Internet protocol and session initiation protocol;
- recommended to support the sharing and control protocols of remote resources, such as binary floor control protocol, remote display protocol and remote frame buffer;
- recommended to support protocols based on media transmission delay, such as real-time transport protocol, secure real-time transport protocol and real-time streaming protocol;
- the service grid capability can optionally be incorporated to provide microservice network call function.

8.1.3 Application programming interface gateway

An API gateway is required to have the following basic functions.

- Request management: unified access capability is required. As the entry point of calling services, it is required to support the functions of service calling, load balancing and protocol conversion. The gateway is required to connect with the service calls of various frontend channels such as external microservices, apps and applets.
- Access control: basic access control capabilities are required, such as black-and-white list control.
- Protocol conversion: conversion of the external requested protocol into the internal service-supported protocol is required. Support for asynchronous and synchronous protocols is required. The ability to shield asynchronous processes should be provided.
- Routing strategy: route frontend requests to different implementations of services in the gateway layer.

An API gateway is recommended to have the following expansion functions:

- gateway collaboration: is recommended to support multi-registry support and to provide the switching management function of multi-registry clusters under distributed gateway clusters;
- filter management: is required to support the non-intrusive hot plug expansion of gateway functions in the form of filters – third party plug-in extension features and a dynamic loading mechanism are recommended to provide open platform services for third parties.

An API gateway is also required to have the following management and maintenance functions for the gateway.

- Service governance functions are required, such as a gateway cluster management panel, log query and early warning management. It is recommended that visual management be supported and real-time routing topology and gateway cluster topology display function be provided.
- A format check can optionally be carried out: for the mandatory constraints in the interface, whether to intercept calls that do not meet the specification in advance can optionally be checked at the gateway layer. The template engine and other technical means can optionally be used to configure and verify the inspection rules.
- Data conversion can optionally be supported: convert the returned data into different formats such as JavaScript object notation or extensible markup language (XML); use the data template engine.
- Backend for frontend (BFF) mode can optionally be supported: BFF can be used as a plug-in of the service gateway.

8.1.4 Load balancing

The following load-balancing capabilities for microservices are required to ensure high availability of microservices:

- load-balancing capability of backend microservices;
- common load-balancing strategies, such as random, polling, weighted polling, IP hashing and minimum connections;
- support for the load-balancing capability of client-side microservices is recommended.

The following capabilities of the network are required to ensure cooperation in load balancing between the network and hardware:

- L4 layer load balancing;
- recommended support for L3 load balancing;
- can optionally support L2 layer load balancing.

8.1.5 Service fault tolerance

The following service fault-tolerant means are required to ensure the availability of the system:

- fault-tolerant mode of circuit breaker in three modes: closed, open and semi-open;
- the system to set a static or dynamic threshold for the number of services processed, and the requests exceeding this threshold shall be directly rejected;
- manual current limiting, degrading and fusing of a single microservice;
- recommended to degrade some unimportant services, and the maximum access limit or diversion of specific microservices should be automatically managed through flow restriction;
- recommended to support automatic microservices fault tolerance processing, including elegant online and offline capability, and automatic removal of instance exceptions;
- recommended to support grayscale release capability for the full microservice call chain.

NOTE – AB test is a grayscale distribution, so that some users continue to use A, other users start with B, if the user has no objections to B, then gradually expand the scope so that all users are migrated to B. Grayscale release can ensure the stability of the whole system, problems can be found and can be adjusted at the initial grayscale level to ensure its impact.

8.1.6 Configuration centre

The following functions are required.

- Static and dynamic configuration: once the system is initialized, it is recommended to dynamically modify load configuration information without restarting; it is recommended to dynamically configure it through the user interface (UI) management interface.
- Provision of control and management functions is recommended: the configuration of the distributed system is centrally managed with a simple and easy-to-use interface. Basic permission management is supported where the configuration information can be edited, published and roll backed, the historical versions of the configuration information can be viewed.
- Support for a variety of configuration modes is recommended: it is required to support the configuration management for file types such as attribute and XML; support for database management of configuration resources is recommended; it can optionally support functions such as configuration changes and change callbacks through an API using programming mode.

8.1.7 Service security

The following means are required:

- client requests are required to be received and managed securely through the API gateway;
- authentication certification based on hypertext transfer protocol (HTTP) or hypertext transfer protocol – secure, session and token is required;
- recommended to store identity authentication in a special authorization server to provide access rights for microservices;
- recommended to support user-defined authentication plug-ins;
- recommended to support multi-factor authentication, including service authentication capabilities based on black and white lists;
- encryption is recommended for sensitive message content; it is optional to provide tampering verification capability for important messages.

8.1.8 Service log

The following capabilities are required:

- record basic log information of the system: including the basic monitoring information of the server, network, storage, middleware and database;
- ensure the integrity of the log: in the log, the level of the log event, the class name and method name of the event, the relevant stack trace information and exception messages are required to be completely stored and recorded;
- ensure the uniqueness of the recorded identifier (ID): the application performance of the system and each request in the business log record are required to have a unique ID (which can be composed of fields automatically generated by the system and multiple fields with business significance);
- generate an alarm immediately in response to serious errors: when a serious error occurs in the processing process and continually affects the business and requires manual intervention, a direct log alarm is required;
- recommended to have the function of customizing the business log content of the system: the format and output content of the log file can be customized according to the business rules of different business systems;

- recommended to support flexible log retrieval capabilities: support keyword and regular expression search, and searching logs by call chain ID.

8.2 Extended functional area of microservice governance

The basic technology subdomain service layer is the derivation and expansion of the basic capabilities in the microservice technology architecture. It includes some advanced functions of large-scale distributed systems.

8.2.1 Distributed transaction management

The following capabilities are required:

- basic transaction management capability based on segmented submission mechanism, such as the extended architecture protocol;
- basic transaction management capabilities based on compensation mechanism, such as "try, confirm, cancel" mode (for retry compensation transaction management) and Saga mode;
- transaction management capability based on a reliable message queue mechanism can optionally be provided;
- other advanced distributed transaction management capabilities can optionally be supported.

8.2.2 Call chain management

The following capabilities are required:

- traceability of any microservice invocation, mainly to provide comprehensive tracking and management capabilities of the microservice call chain;
- for each call chain, basic information such as microservice name, microservice start time and end time is required, references to other call chains is required, and the customized identification of microservice is recommended;
- management ability of data collection, data persistence and data display of tracking data;
- recommended to support linkage between the call chain and the log, so the call log can be automatically associated in the service call, providing quick troubleshoot.

8.2.3 Third party microservice module access

The following capabilities are required:

- container cloud platform and image warehouse to support the uploading of third party microservice module images;
- a running environment that supports the mainstream microservice module development language;
- a third party microservice module interface conforming to the specification to realize business access and system access.

8.2.4 Framework expansion capacity management

The following framework expansion methods are required:

- SPI plug-in mode is recommended: suitable for situations with high performance requirements or complex expansion system;
- filter mode: suitable for simple expansion requirements and can be dynamically added or deleted.

8.2.5 Message-oriented middleware

The following capabilities are required:

- a persistence mechanism to retain message data;

– message distribution mechanisms such as distributed subscription, polling distribution and message pull back;

– distributed message processing capability;

– recommended to support open source message system;

– message protocols for the Internet of things can optionally be supported;

– using HTTP as the messaging protocol is prohibited.

8.2.6 Metrics measurement management

Monitoring of metrics acquisition, log collection and other methods is required to provide the following system operation status indicators:

– key performance parameters: the number of calls, response time and response code status of the microservice API; key indicators in network transmission: delay, packet loss, jittery, bandwidth and other indicators; coding, code rate, frame rate, resolution, encryption or not in multimedia applications; data query or update times, query or update duration, number of transactions per second, etc.;

– key event parameters: event name and occurrence time in the business process; the status of the triggered microservice sending or receiving messages;

– recommended to support key path parameters: service nodes passing through the business process and services participating in the process; the address of the node and the port number of the microservice; network topology between nodes, number of connections and hops;

– recommended to provide business-level identification information, business serial number, user identification information, etc.

8.2.7 Serverless function management

The following functions are recommended:

– the lifecycle management of functions and the trigger execution of functions: including the launch online, running, deletion, updating and other operations of functions;

– function cold start: the start process needs to consider resource scheduling, container start, function code download, function start, function initialization and function processing call requests;

– function elastic scaling: it should make elastic decisions according to the resource load of the application or according to the requested traffic;

– for stateful functions, it is recommended that programme state management functions be provided: the operations of state management include state generation, state access, state operation, state deletion and state recycling.

8.2.8 Service mesh management

The service mesh style of microservice management can optionally be integrated into MMA. It should be able to decouple from other infrastructural resources though. The following functionalities are required.

a) A high-performance communication agent (called sidecar) which is responsible for the communication between various services. It is required to have the ability to manage sidecars, support viewing and managing sidecar instances in the current workspace, and provide sidecar status, injection time and other information.

b) Function of control surface to manage configuration and monitor data, including the following.

  i) Telemetry recording service, process telemetry records and transmission of the request indicators generated by the agent sidecar to the designated backend.

ii) Policy implementation services, mainly responsible for the implementation of cluster management strategies.

iii) Authentication services: authentication between services, users and services to achieve access control. It should include a public secret key system to provide the generation, rotation, revocation and end-to-end verification services of client TLS certificates for each microservice in the service grid.

iv) Recommended to support multi-programming language access.

v) Recommended to provide a user-defined resource controller to handle the change of user-defined resources and distribute content to other relevant services in the service network.

vi) Recommended to provide guidance services. For custom resources in the service mesh, it is required to provide standardized configuration and orchestration management to guide them into the sidecar container of the agent.

c) The function of the data plane should be able to handle the network connection between microservices in the service grid, receive the routing and policy rules of microservices from the control plane, and report the results of connection processing to the control plane.

8.2.9 Multi-modal framework compatibility management

The following functions are recommended to provide the compatibility of managing heterogeneous microservices across microservice frameworks.

– It is required to support unified service management to deal with access requests of backend services, so as to improve the capabilities of service governance, high-speed access, etc., which is called a gateway bridge.

– Use some interface description language to define the API of services, so as to help realize the maximum cross-platform microservice API calls.

– Runtime can optionally be used to realize the core business functions of microservices, and the runtime can automatically adapt to the changes of the execution environment. The runtime here refers to the virtual machine at the process level under the semantics of cloud computing.

8.2.10 Fitness function management

The fitness function is recommended to support the following management tools:

– measurement tools: tools for measuring nodes and clusters in complex networks;

– simulation tools: tools for simulating and calculating measurement parameters.

8.2.11 Other extended capacity management

The extended management can optionally further include the following capacities:

– for large distributed systems, it is recommended to support high availability, high concurrency (elastic scalability), high reliability (security compliance) and other expansion capabilities;

– for multi-cloud and multi-data centres, it is recommended to support multi-centre business docking and traffic scheduling capabilities;

– other expansion capabilities such as webhook can optionally be supported.

9 Media business service layer

9.1 Media commonality support function area

The following audiovisual media common service capabilities are required:

- media file management: addition, loading, file name modification, deletion, querying, transmission, etc. of media files;
- audio and video content processing: encoding services, compression services, push/receive services, protocol/format conversion, audio control, frame extraction services, streaming media slicing services, streaming media live broadcast acceleration, artificial intelligence (AI) resolution services, cloud editing services, etc. for multimedia streams;
- interactive management: support short message service gateway, comments, etc.; should support barrage services;
- user attribute management: IP attribution service, accessibility service, management of minors, etc.

The following audiovisual media common service capabilities are recommended:

- digital rights management: include basic capabilities such as copyright certification archive, copyright confirmation registration and copyright asset management; it is recommended that infringement judgment, evidence fixation and other rights protection capabilities be included; it can optionally provide a mechanism for early warning and risk control;
- media AI service: provide full lifecycle management services for audio and video AI models, and realize the "model-as-a-service" capability for the generation and processing of audio and video content;
- resources unified caching management: provide local caching, distributed caching, reverse proxy caching, browser caching and content distribution network caching capabilities for media files; strategies such as hotspot caching and multi-level caching should be provided;
- audiovisual synchronization: time synchronization of audio and video content such as audio and video streams, and files;
- other common services: such as news cross-checking and fake news detection, fake voice and pseudoface detection, and other services.

10.1 Special media business function area

10.1.1 Insight and planning domain

The following capabilities are recommended:

- cue aggregation: aggregate cues over multiple channels;
- hot-spot analysis of public opinion: public opinion hot spots can be analysed by personalizing time, events, people, regions, etc.;
- topic selection planning: carry out unified topic selection planning to realize topic creation and editing, topic selection screening, topic selection review and distribution, combined reporting, etc.

10.1.2 Production and broadcasting domain

The following capabilities are recommended:

- content aggregation: aggregate text, pictures, audio, video and other content over multiple channels;
- content production: produce audiovisual media content for multiple distribution channels;

- content broadcasting: broadcast audiovisual media programs for multiple distribution channels.

10.1.3 Transmission and distribution domain

The following capabilities are recommended:

- channel distribution: transmit and distribute programs to radio and television channels;
- client publishing: release audiovisual media content to client application;
- website publishing: publish audiovisual media content for the website.

10.1.4 Operation domain

The following capabilities are recommended:

- operation and management of platform users and various software applications;
- unified portal to support unified authentication and login of users;
- services for users with different organizations and needs in a multi-tenant manner, meaning that permissions, contents and information among tenants are independent of each other;
- users with personalized and programmable service interfaces, and users can independently arrange the presentation sequence and layout of application services according to their work needs;
- unified management of users with the ability to provide user information to all application systems;
- unified management of application services, including the functions of turning applications on and off, billing of platform access rights, etc.;
- functions for creation, modification, publication and deletion of announcements.

10.1.5 Content supervision domain

The following capabilities are recommended:

- ability to detect the quality problems of audio and video media, such as comprehensive evaluation of jitter ghosting, blur, black screen, white screen, noise, mosaic, snowflake, static frame, frame level, frame skipping, picture quality, audio and video synchronization, audio freezing, muteness, no track, audio inversion, audio error, and audio and video quality;
- intelligent review ability, such as the subjective quality evaluation system for on-demand video;
- scenery recognition, face feature extraction, star recognition, image material generation, etc.;
- caption optical character recognition, caption detection, etc.;
- it is optional to provide advertising induction review and advertisement placement correction services.

10.1.6 Security domain

The following capabilities are recommended:

- login authentication: support the identification and authentication of the logged in user;
- access control: restrict and manage access to users or other systems through authorization, authentication, access control lists, roles, single sign-on, two-factor authentication;
- security audits: support the audit of important user behaviours and important security events;
- data security: support the confidentiality of important data during transmission and storage through cryptographic technology;

- signature verification: support the integrity of signature verification of important data during transmission and storage through verification or cryptography;
- threat monitoring: support continuous detection of malicious and unauthorized behaviour, identify various potential threats and output analysis results.

10.1.7 Monitoring operation and maintenance domain

The following capabilities are recommended:

- service operation status monitoring: support real-time monitoring of the operation status of business systems, applications, network equipment, etc.;
- service operation performance monitoring: support the monitoring and analysis of various performance indicators such as business processing and data transmission of business systems;
- business resource utilization monitoring: support the monitoring and analysis of resources (such as CPU, memory, disk space, bandwidth, etc.) used by business systems;
- user visits: support statistics on the number of users who visit pages, resources or functions within a certain time range;
- automated service inspection: support inspection of security policies (such as alarms, monitoring and access control) of business systems, and automatic discovery and diagnosis of business problems.

11 Platform service layer

11.1 Business application programming interface lifecycle management

The following capabilities are recommended:

- required to have API code development capability: tools can be used to implement API code, such that auxiliary files can be generated automatically;
- required to have API code test capability: it is required to have manual testing capability, and it is recommended to have automatic testing capability;
- required to have API publishing: authorization management, permission key management, etc.;
- API design capability: the caller of API needs to be fully verified and provide feedback;
- API deployment: different operating environments can be chosen such as cloud, local server and local cluster;
- operation and maintenance of API: key performance indicators are quantitatively measured and the health of enterprise operation is monitored.

11.2 Process engine service

The following service capabilities are recommended in business activity analysis, business process design, business process simulation, business process operation, business process monitoring and other processes:

- required to flexibly combine the underlying basic services, define self-service processes and expand service capabilities;
- required to monitor and manage the status of process instances through a system management and monitoring interface, including user management, role management, log management, resource control and process monitoring;
- graphical and visual process definition tools;
- tool calls process application data;

- adopt two methods to execute the tool program, including HTTP interface and message queue;
- workflow security mechanism, including authentication, authorization, access control, audit, data confidentiality, data integrity, anti-denial and security management;
- data query, multi-dimensional statistics, report presentation and other functions through web pages or standard API interfaces;
- can optionally provide workflow execution services to complete the creation, execution and management of workflow process instances.

11.3 Rule engine service

The following rule engine service capabilities are recommended:

- required to support script rule sets, and recommended to support wizard rule sets;
- required to support general decision rule table and cross-decision rule table;
- required to support ordinary scorecards. recommended to support complex scorecard functions;
- decision trees, i.e., business rules are displayed in a tree form.

11.4 Microservice trading management

The following capabilities are recommended:

- required to provide a multi-tenant environment for microservice management;
- tenants are required to have the basic ability to create new service instances, modify microservice instances, renew service instances and unsubscribe from service instances;
- required to provide unified subscription management capabilities for microservices;
- required to provide unified accounting management capabilities for microservices;
- a unified pricing strategy for microservices;
- the ability of unified bill management of microservices, such as generating real-time bills;
- the ability of unified payment management of microservices.

11.5 Microservice quality management

The quality management of microservices is recommended to provide the following capabilities:

- required to provide the ability of microservice contract testing;
- required to provide the ability of microservice performance testing;
- the ability of microservice virtualization.

11.6 Other platform management

It is optional to support common object request agent middleware, common language runtime and database connection middleware to further expand the capabilities of this layer.

12 Application integration layer

12.1 General

In MMA, any software, systems or platform services that can realize some specific functions of the system in the form of microservices and are composed of process arrangement and application integration are called applications. The construction of audiovisual media software applications is recommended to be based on the business pool of microservice components in the media business service layer, and then by relying on the platform service capability layer for collaboration, the application integration capability of this layer is assembled.

12.2 Open application model management

During application construction, the following functions are recommended:

- required to divide applications into components to form multiple application components: these components can call relevant microservices to build their own functions;
- required to have an application deployment configuration file to describe the relationship between components and the called microservices;
- application operation and maintenance feature information for application in the specific deployment environment.

12.3 Application adaptation management

Application adaptation management is recommended to provide the following capabilities:

- required to have an application interface adaptation centre to provide API hosting and adaptation services;
- an application connector plug-in framework model to support plug-in expansion based on basic connectors;
- it is optional to have the ability of intermediate language, and make it possible for applications in different fields to interact by defining routes and intermediate rules.

12.4 Low code development platform

The following capabilities are recommended to provide:

- required to provide a visual designer, which can visually program the basic components of common application construction such as forms, portals, processes and rules;
- required to have the ability for visual construction: it can build and reconstruct key components of business modelling, process design, UI page design and other applications through drag and drop and other visual operations;
- required to have the ability to visually display the basic components of software application;
- the ability to configure and manage the basic components of software applications such as code base and microservice components;
- one click release, visual application status supervision and other auxiliary capabilities;
- the ability to operate through the web UI besides the desktop UI.

12.5 Application portal management

Application portal management is recommended to provide the following capabilities:

- required to have the ability to display the selected content of different applications to a specific UI window;
- required to have the ability of unified layout and management of all kinds of UI windows;
- the function of personalized customization of a single application;
- the ability of unified permission management for management applications, such as single sign-on.

12.6 Micro frontend management

Micro frontend management is recommended to provide the following capabilities:

- required to have the ability to arrange and manage UI components;

- required to separate the control and display of frontend components;
- sandbox operation environment for frontend applications.

12.7 Other integration capabilities

It is optional to support audiovisual media applications such as broadcast system, production system and media resource management.

13 Development and operations engineering

13.1 General

In order to cope with the construction and operation, and maintenance of complex software systems and prevent the gradual corruption of large-scale systems, MMA also needs to cooperate with development and operations (DevOps). This clause specifies the competencies that shall be possessed by continuous integration and continuous deployment and anti-corruption management in DevOps.

13.2 Continuous integration and delivery management

The following capabilities are required:

- establishing pipelines: work such as code integration and packaging, and publishing is connected through a series of (open-source) tools to form an automated pipeline;
- submitting code: the newly submitted codes are automatically edited and updated; sometimes automatic detection of code quality is included;
- submitting testing: covering contract test, unit test, integration test and other capabilities;
- publishing deployment: codes are changed and edited into different software packages, and automatically released to different environments such as testing, integration and production.

13.3 Anti-corrosion management

Anti-corrosion management is required to provide the following capabilities.

- Strangler pattern: smooth transition between the old and new systems should be ensured by gradual replacement rather than one-time replacement.
- Recommended to provide sidecar pattern: auxiliary components of the application are deployed as separate containers or processes to provide isolation and encapsulation.
- It is optional to provide bulkhead pattern: key resources of each workload or service are isolated, such as connection pool, memory and CPU. The use of bulkheads avoids the scenario that a single workload (or service) consumes all resources, resulting in the failure of other services. This pattern mainly increases the flexibility of the system by preventing cascading failures caused by one service.

Other corrosion prevention patterns are recommended:

- unified internal file migration service for storage, where files are shared through hard links without the need for the true copy and migration of physical files;
- register all file information on the platform service layer, which is uniformly managed by the platform service layer, including the management of lifecycle and logical storage space;
- it is optional to realize data migration and transmission within the platform and between the platform and external systems; it is optional to support the migration of specified files to another business network, including creating a new migration task, querying migration tasks, suspending and resuming tasks, cancelling tasks and modifying task priorities.

Appendix I

Description of third party microservice module access

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

I.1 Principle

The third party microservice module access can effectively expand the capabilities of the microservice platform, break down technical barriers, and endow the platform with the characteristics of openness, sharing, and standardization. This function is shown in the API gateway in the architecture diagram, but it belongs to the recommended advanced function, so it logically belongs to the service of this layer.

The third party microservice module access consists of three layers, as shown in Figure I.1. The bottom layer is physical access, whose purpose is to allow the code of the third party microservice module to run on the platform. The middle layer is access of business, whose purpose is to allow the business (function) of the third party module to be used on the platform. The upper layer is access of system, whose purpose is to allow the business of third party microservice modules to be invoked in the orchestration system.

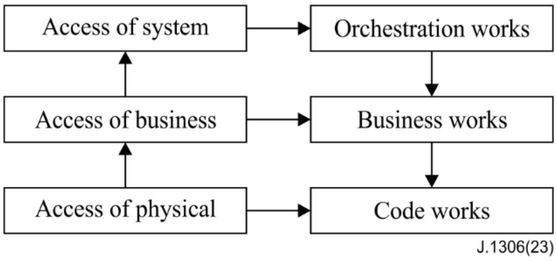

Figure I.1 – Access of three layers

I.2 Architecture

In order to access the three layers, the technology architecture of third party microservice module access is shown in Figure I.2.

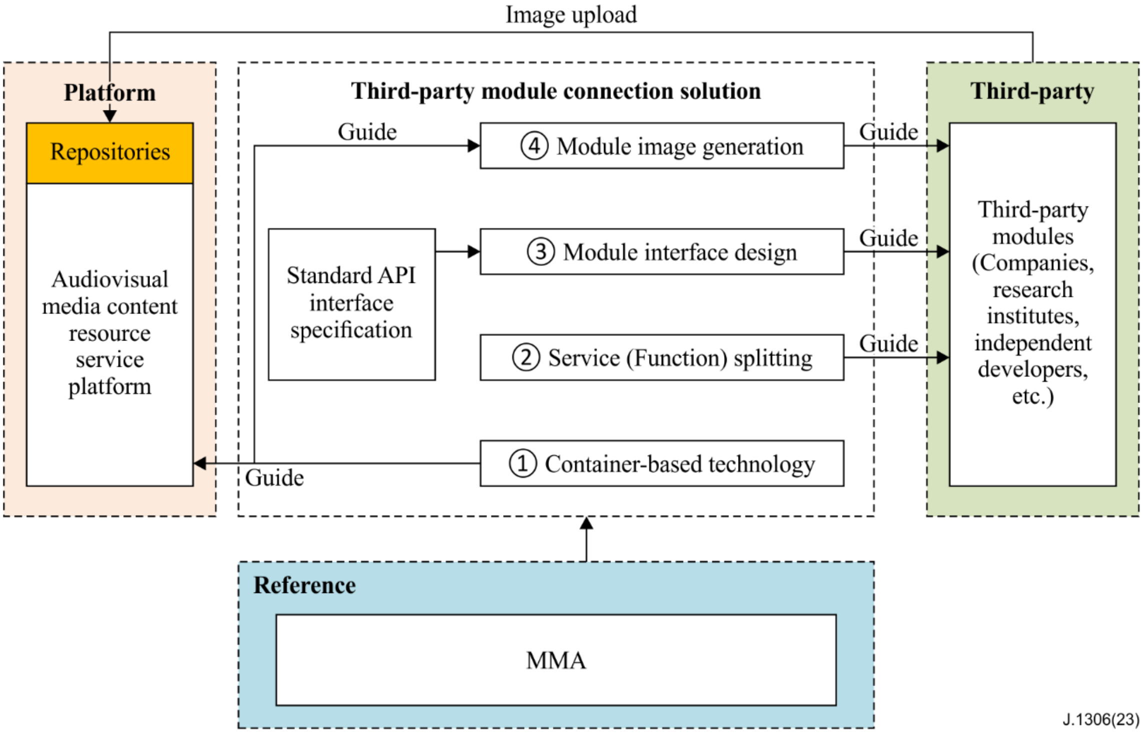
Image upload
Platform
Repositories
Audiovisual media content resource service platform
Third-party module connection solution
Guide
④ Module image generation
Guide
Standard API interface specification
③ Module interface design
Guide
② Service (Function) splitting
Guide
① Container-based technology
Guide
Third-party
Third-party modules (Companies, research institutes, independent developers, etc.)
Reference
MMA
J.1306(23)

Figure I.2 – Access technology architecture of third party microservice module

After MMA is referred to, it is clear that container cloud and image repository are the basis for third party microservice module access. Based on sub-domains of the seven functions of audiovisual media, this architecture provides a classification scheme for audiovisual media micro-services, formulates standardized micro-service module interface specifications, and provides guidance for third party microservice module development teams. The microservice module mirror production technology is studied, and the third-party module development teams are guided to package the microservice module codes and operating environment and upload them to the platform mirror warehouse, so that the microservices can be deployed and run across platforms.

Bibliography

[b-ITU-T H.705.1] Recommendation ITU-T H.705.1 (2023), *Layered specification for the IPTV service platform functional architecture based on open service capabilities*.

[b-ITU-T J.1301] Recommendation ITU-T J.1301 (2021), *Specification of a cloud-based converged media service to support Internet protocol and broadcast cable television – Requirements*.

[b-ITU-T J.1303] Recommendation ITU-T J.1303 (2022), *Specification of a cloud-based converged media service to support internet protocol and broadcast cable television – System specification on collaboration between production media cloud and cable service cloud*.

[b-ITU-T X.1193] Recommendation ITU-T X.1193 (2011), *Key management framework for secure Internet protocol television (IPTV) services*.

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

[b-ISO 19297-1] International Standard ISO 19297-1:2019, *Intelligent transport systems – Shareable geospatial databases for ITS applications – Part 1: Framework.*

[b-ISO/IEC TS 23167] Technical Specification ISO/IEC TS 23167:2020, *Information technology – Cloud computing – Common technologies and techniques*.

[b-ISO/TR 11219] Technical Report ISO/TR 11219:2012, *Information and documentation – Qualitative conditions and basic statistics for library buildings – Space, function and design*.

SERIES OF ITU-T RECOMMENDATIONS

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

Published in Switzerland
Geneva, 2023