

ICS 33.040.40

M 32



中华人民共和国通信行业标准

YD/T 2260-2011

IPTV机顶盒中间件技术要求

STB middleware system architecture

2011-06-01 发布

2011-06-01 实施

中华人民共和国工业和信息化部 发布

目 次

前 言..... II

1 范围..... 1

2 术语和定义..... 1

3 缩略语..... 1

4 IPTV系统总体中间件架构..... 3

5 IPTV终端中间件架构..... 4

 5.1 设备驱动及资源系统..... 4

 5.2 IPTV终端中间件..... 4

 5.3 应用层..... 6

6 中间件对外接口..... 6

 6.1 中间件API接口..... 6

 6.2 资源抽象层..... 9

7 设备驱动及资源系统..... 10

 7.1 设备驱动..... 10

 7.2 资源层软件架构..... 11

 7.3 运行和通信机制..... 12

8 插件管理..... 12

 8.1 插件的提供方式..... 12

 8.2 安全模式..... 13

9 中间件的发行认证机制..... 13

 9.1 中间件的发行注册认证流程..... 13

附录A（规范性附录） 中间件软件的数字签名..... 14

前 言

本标准是IPTV业务系统系列标准之一。该系列标准的结构和名称预计如下：

1. YD/T 1654-2007 《IPTV 业务需求》
2. IPTV 用户场景要求
3. YD/T 1823-2008 《IPTV 业务系统总体技术要求》
4. YD/T 1697-2007 《IPTV 内容运营平台与业务运营平台接口技术要求》
5. YD/T 2015-2009 《IPTV 业务管理系统体系架构》
6. YD/T 1655-2007 《IPTV 机顶盒技术要求》
7. YD/T 2017-2009 《IPTV 机顶盒测试方法》
8. YD/T 1696-2007 《机顶盒与 IPTV 平台接口技术要求》
9. YD/T 1824-2008 《IPTV 终端管理系统体系架构》
10. YD/T 1825-2008 《IPTV 终端机卡分离技术要求》
11. IPTV 安全体系架构
12. YD/T 1920-2009 《IPTV 承载网络体系架构》
13. YD/T 1695-2007 《IPTV 对接入网络的技术要求（第一阶段）》
14. IPTV 媒体交付系统技术要求 场景和需求
15. IPTV 媒体交付系统技术要求 体系架构
16. IPTV 系统的内容管理系统
17. IPTV 内容安全架构
18. YD/T 2016.1-2009 《IPTV 运维支撑管理接口技术要求》

——第 1 部分：业务系统

——第 2 部分：承载网络

——第 3 部分：终端

本标准由中国通信标准化协会提出并归口。

本标准起草单位：UT斯达康（重庆）通信有限公司、中国电信集团公司、中国联通通信网络集团有限公司、华为技术有限公司。

本标准主要起草人：张美春、孟建庭、陶孟华、候伟奕。

IPTV机顶盒中间件技术要求

1 范围

本标准规定了 IPTV 机顶盒中间件的总体架构、功能要求、软件系统、中间件对外接口、中间件与系统的相关协议。

本标准适用于IPTV机顶盒中间件。

2 术语和定义

下列术语和定义适用于本文件：

2.1

IPTV 业务 IPTV Service

通过IP网络提供的，具有交互能力，可以支持组播、点播等多种播放形式的多媒体服务业务。

2.2

IPTV 中间件 IPTV Middleware

介于资源层和应用程序之间的一层软件，它包含一组由多个可以运行在一个或多个设备上且与系统进行交互的功能所构成的服务。

2.3

资源 Resource

终端中一组硬件和软件功能实体的集合，包括内存、USB接口、编解码器等硬件功能和定时器等软件功能。

2.4

资源抽象层 Resource Abstraction Layer

IPTV中间件系统移植到不同的硬件和驱动层平台时所定义的一系列与资源层之间的接口。

3 缩略语

下列缩略语适用于本文件：

BRAS/BAS	Broadband Remote Access Server	宽带远程接入服务器/宽带接入服务器
BSS	Business Support System	业务支撑系统
CDN	Content Distribution Network	内容分发网络
CDR	Call Detail Record	呼叫明细记录
CMS	Content Management System	内容管理系统
CP	Content Provider	内容提供商
CPU	Central Process Unit	中央处理单元
CSS	Cascading Style Sheets	层叠样式表单
DHCP	Dynamic Host Control Protocol	动态主机控制协议
DoS	Denial of Service	拒绝服务
DRM	Digital Rights Management	数字版权管理
DVB	Digital Video Broadcasting	数字视频广播

EJB	Enterprise JavaBean	企业 JavaBean
EPG	Electronic Programmer Guide	电子节目单
FTP	File Transfer Protocol	文件传输协议
GIF	Graphics Interchange Format	图片交换格式
HTML	Hypertext Markup Language	超文本标记语言
HTTP	Hypertext Transfer Protocol	超文本传输协议
HTTPS	Hypertext Transfer Protocol Secure	安全超文本传输协议
IGMP	Internet Group Management Protocol	因特网组管理协议
IPTV	Internet Protocol Television	IP 电视
JAAS	Java Authentication Authorization Service	Java 验证和授权业务
JNDI	Java Named Directory Interface	Java 命名目录接口
MPEG	Moving Picture Experts Group	移动图像专家组
MPEG-1	Moving Picture Experts Group 1	移动图像专家组 1
MPEG-2	Moving Picture Experts Group 2	移动图像专家组 2
MPEG-4	Moving Picture Experts Group 4	移动图像专家组 4
NTP	Network Time Protocol	网络时间协议
NVoD	Near Video on Demand	准 VoD
OCS	Open Content Source	开放内容源
OSS	Operation Support System	运营支持系统
PPV	Pay Per View	按次付费点播
PVC	Perment Virtual Connection	固定虚连接
PVR	Private Video Recorder	个人视频记录器
QoS	Quality of Service	服务质量
ROAP	Rights Object Acquisition Protocol	版权对象获取协议
SDP	Session Description Protocol	会话描述协议
SI	Service Information	业务信息
SOAP	Simple Object Access Protocol	简单对象存储协议,
SP	Service Provider	服务/业务提供者
STB	Set Top Box	机顶盒
TCP	Transmission Control Protocol	传输控制协议
TSTV	Time-shifted Television	延时(时移)电视
URL	Uniform Resource Locator	统一资源定位符
UTF-8	UCS Transformation Format-8	UCS 传输格式 8
VOD	Video On Demand	视频点播
WMV	Windows Media Video	微软视频编码格式
WMv9	Window Media Version 9	窗媒体 v9
XML	Extensible Markup Language	扩展标记语言
XrML	Extensible Rights Markup Language	扩展权限标志语言
XSL	eXtensible Stylesheet Language	可扩展样式表语言

4 IPTV 系统总体中间件架构

IPTV 中间件架构的作用不仅是将应用程序和底层系统资源隔离，另一个重要的作用是实现应用与应用之间的交互。在 IPTV 系统中，中间件模块完成了机顶盒、IPTV 平台系统、应用三者之间交互协作。

图 1 是中间件在系统中的位置。

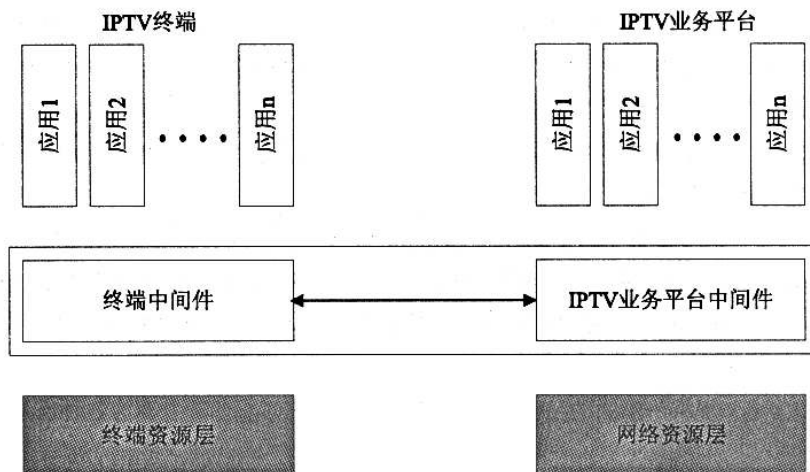


图1 IPTV系统中中间件的位置

1. IPTV终端侧：终端应用通过中间件使用终端的资源，完成终端的业务实现；
2. IPTV平台侧：平台应用功能通过中间件调用平台资源，完成平台业务功能的提供。

IPTV系统总体中间件架构如图2所示，其中中间件模块向上给应用层提供API接口，向下提供可移植层接口，核心部分是中间件的各功能块。

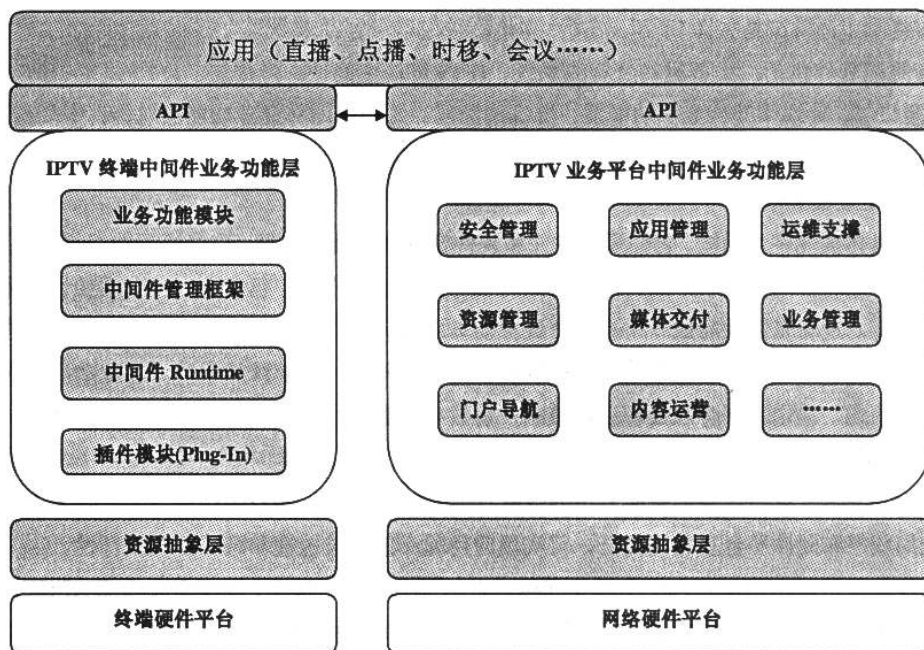


图2 机顶盒中间件在IPTV系统中的位置

中间件系统应该包含三层：

资源抽象层（Resource Abstraction Layer: RAL），使中间件能在一些可选择性的平台上运行，比如服务器、机顶盒和PC客户端等。资源抽象层必须提供一系列可移植层接口功能。

中间件业务功能层（Service Function Layer）是中间件体系中的核心部分，它由一系列业务功能模块构成，这些业务模块可以分布式或集中式地装载在IPTV终端和IPTV服务器中。IPTV终端中间件是指装在IPTV终端的中间件业务功能模块。IPTV业务平台中间件则是指装在各IPTV服务器中的中间件业务功能模块。IPTV终端中间件的业务功能模块可以是IPTV业务平台中间件的客户端（Client）或者独立存在。IPTV终端包括机顶盒、PC机或者手机，本规范主要讨论IPTV机顶盒中间件的体系架构，PC机和手机的中间件不在本规范讨论范围之内。

根据IPTV的架构，IPTV业务平台中间件业务功能层应该包含下列功能模块：内容运营功能、运维支撑功能、门户导航功能、业务管理功能、媒体交付功能、安全管理功能、资源管理功能。IPTV终端中间件业务功能层应该包含下列功能模块：各类中间件业务功能模块、中间件管理框架（Framework）、中间件Runtime、插件模块。

API层（API Layer）提供一系列的API，为业务提供商和第三方服务开发商为各种应用目的开发特定的应用并呈现出来。

- 业务 API，一种业务使能 API 如机顶盒认证、获取用户帐务信息等；
- 功能 API，主要是为了特定的功能应用如媒体播放控制、短消息发送等；
- 编程 API，主要是终端设备开发的 SDK 所用，如内存分配等；
- 基于这些 API，应该支持各种操作系统（Windows, Windows CE, or Linux），通用协议（IP, SOAP/XML），和各种编程接口（C/C++, Java, PHP）。

5 IPTV 终端中间件架构

IPTV终端中间件架构如图3所示。

IPTV终端软件包括：设备驱动及资源系统、IPTV终端中间件、应用层。IPTV终端中间件包括：中间件API接口层、终端中间件业务功能层、资源抽象层。终端中间件业务功能层包括：中间件管理框架、中间件业务功能模块库、中间件RunTime、插件模块（Plug-In）。

5.1 设备驱动及资源系统

资源层具体实现许多功能，它们通过软件和硬件资源来表达和实现。这些资源的组织形式并不固定，不同的设计方案有不同的组织形式。这种模式可以考虑在整个平台中包含不止一个硬件设备和功能。

抽象的看，逻辑资源映射成一个或多个硬件实体并没有什么不同，重要的是资源是提供给应用的。一个应用程序应该能够处理所有本地连接的资源，就像它们是一个实体的组件。在驱动层中必须装有一个实时操作系统，决定其任务和线程调度策略，协调整个系统中各个线程的运行。

5.2 IPTV 终端中间件

应用程序并不直接调用底层资源。中间件软件平台软件层为这些资源提供了一个抽象层，该中间抽象层将应用程序和硬件平台互相隔离开，可实现应用程序的可移植性和跨平台可执行性。

中间件软件平台中包含有一个中间件框架，它负责启动应用程序、插件、中间件API库，管理所有应用程序、插件、中间件API库的生命周期，包括彼此之间的相互操作，便于各种应用程序能够协调地执行。



图 3 IPTV 终端中间件架构

5.2.1 中间件API层

中间件API层向应用层开放一个让应用开发商迅速创建应用业务的应用程序接口。这些接口包含两部分：中间件功能接口和中间件管理框架接口。中间件功能接口覆盖了终端及终端和系统进行交互的各种功能，包括媒体传输和控制功能、升级下载服务功能、安全和认证功能、元数据接入管理功能等。中间件管理框架接口对应用层使用中间件功能接口提供必需的安全、管理支持。

中间件API是和终端平台及网络无关的，基于该API开发的应用能在不同的支持该API标准的网络上运行，也可以在支持该接口的不同终端平台上运行。

5.2.2 终端中间件业务功能层

终端中间件业务功能层实现中间件管理框架、中间件的核心模块库、中间件RunTime及插件等功能。

5.2.2.1 中间件管理框架

中间件管理框架为中间件业务功能层及应用层调用中间件功能提供一些必要的外围支撑能力，如服务注册与发现、认证鉴权、服务订购、完整性管理等。中间件管理框架的作用是确保中间件功能接口的开放性、安全性、一致性和可管理性，保证了中间件业务功能层的有序运行。

5.2.2.2 中间件业务功能模块库

中间件模板库实现了对终端和系统交互的能力的抽象，它允许应用程序访问网络的各种能力，如媒体控制、升级下载、用户认证、消息服务等。

5.2.2.3 中间件RunTime

中间件RunTime实现各种协议及各种基本的中间件内核程序。

5.2.2.4 插件模块 (Plug-In)

插件模块允许终端灵活引入各种特有的服务程序，以支持应用层调用它和系统或者增值服务器交互实现各种应用。

5.2.3 资源抽象层

资源抽象层包括系统抽象层和硬件抽象层接口。资源抽象层的目的是将具体的硬件平台实现与中间件内部模块设计隔离开来，使得同一套中间件软件能够平滑的移植到不同的硬件平台。

5.3 应用层

应用程序层是为用户提供IPTV的各种业务、基于机顶盒的各种应用。应用程序分成以下几大类：

- 浏览器及 B/S 方式的各种应用。浏览器又分两种情况：通用浏览器及定制的浏览器。EPG 的呈现可以采用通用浏览器，也可以采用定制的浏览器。增值应用可以基于通用浏览器，采用 B/S 方式开发。通过对浏览器提出要求，如支持 JavaScript、Flash 插件，能通过插件方式调用 IPTV 系统的消息能力等，可以丰富 B/S 方式应用的表现能力。

- C/S 方式的应用程序。C/S 方式的应用程序直接与服务器端交互，实现各种应用，适合于实现无法通过 B/S 方式实现的游戏、证券等应用。

- 本地应用程序。不需与服务器端交互的本地应用，如：本机游戏、用户收藏夹、本地通信录等。

6 中间件对外接口

6.1 中间件 API 接口

中间件通过 API 接口向上层应用提供服务。中间件软件平台包含资源管理、安全和认证管理、初始化及下载服务、升级下载服务、媒体/RTSP 服务器、DRM/CA、用户服务、通信控制、终端管理、元数据管理、Internet 接入等。其中通信控制、资源管理、终端管理、和 Internet 接入为可选组件。双模机顶盒的中间件软件平台基于同样的中间件接口提供 DVB 中间件功能。

6.1.1 初始化及下载服务

初始化及下载服务模块是机顶盒上电运行的第一个软件模块，它负责初始化机顶盒系统、准备运行环境及对机顶盒进行终端。它主要的功能应该包含：

1. 对机顶盒进行初始配置；
2. 系统初始化，包含应用管理初始化、启动 DRM 等相关任务、进行用户认证处理等；
3. 网络链接初始化，包括获取相关网络参数、配置接入模式等等；
4. 系统资源初始化，包含内存的预分配、创建定时器、I/O 设备及 A/V 编解码器的初始化准备；
5. 和其他中间件模块协同工作，如和升级下载模块一起完成软件版本检查及实现软件升级。和安

全认证模块一起完成用户的认证流程；

6. 对机顶盒进行终端。

6.1.2 安全和认证管理

安全和认证管理负责整个系统的安全机制，包括应用程序的鉴权和安全策略、用户认证、网络安全等。当机顶盒下载应用程序时，安全管理器必须对应用程序进行鉴权。运营商对应用程序进行数字签名，安全管理器将下载的应用程序进行数字签名的检查，确保应用程序没有被篡改。没有进行数字签名的应用程序只能具备基本的权限。安全认证管理模块应该包含以下功能：

1. 用户认证及相关信息管理；
2. 应用业务授权；
3. 软件升级和下载时进行认证；
4. 网络安全策略管理；
5. 密钥、口令字及注册信息管理；
6. 家长控制相关信息管理；
7. 内容购买信息管理。

6.1.3 升级下载服务

升级下载服务负责机顶盒系统平台及应用软件的动态版本检查和更新服务，它完成应用程序的动态下载和应用程序所需要的一些 IPTV 特有数据的下载，如电子节目单的下载。升级下载服务模块主要完成下列功能：

1. 应用软件的升级和下载；
2. 中间件层软件的升级和下载；
3. 平台资源的升级和下载；
4. 应用层数据及其他配置数据的升级和下载；
5. 和安全认证模块一起实现软件和数据的合法性检查。

6.1.4 媒体、RTSP 服务

媒体、RTSP 服务管理负责管理媒体流、媒体呈现、媒体存储。对媒体流的控制包含单播、多播媒体流的媒体播放控制，也包括对针对 IPTV 服务的事件进行管理，如快进、快退、暂停、PVR 录制及直播电视热键等。媒体、RTSP 服务模块包含以下主要功能：

1. 媒体流管理，如媒体流会话的建立、VOD 控制、多播、单播、时移等；
2. 每天解码管理；
3. 媒体流呈现和控制模式管理，如播放、停止、暂停等；
4. 字幕呈现；
5. DRM 触发；
6. 媒体缓存管理；
7. PVR 和存储管理。

6.1.5 DRM

DRM 负责处理权限控制消息、权限管理消息、它提供独立的接口。DRM 的接口提供给应用程序一个高级接口来访问条件接入系统，而底层的条件接入系统在处理例如权利控制消息和权利管理消息时，

并没有暴露给上一层的应用程序。DRM 模块屏蔽了不同的 DRM 系统的差异性。DRM 模块的主要功能包含：

1. 许可证管理；
2. 权限管理；
3. 密钥管理；
4. 媒体流和数据流的解密。

6.1.6 元数据管理

元数据管理负责元数据的接入、元数据呈现及业务选择管理。元数据管理主要包含以下功能：

1. 接入元数据系统；
2. 元数据分发系统；
3. 元数据呈现，如节目单和事件目录；
4. 业务选择管理。

6.1.7 用户服务

用户服务模块负责终端用户的交互、分发相应的事件给应用层。用户服务模块的主要功能包含：

1. 通过键盘、鼠标、遥控器和终端用户进行交互；
2. 接收和分发终端用户的事件给应用层；
3. 运营商发送通知给终端用户；
4. 业务运营商的 Logo 显示。

6.1.8 终端管理

终端管理模块提供终端的管理功能，包括配置管理、性能管理、安全管理等。终端管理模块主要的功能包括：

1. 远程控制管理；
2. 日志管理；
3. 软件版本管理；
4. 系统参数配置，如服务器地址配置等；
5. 接入模式和相关参数配置；
6. 媒体服务参数配置如编解码的配置等；
7. 用户参数配置如接入账号配置、业务账号配置等。

6.1.9 资源管理

资源管理器负责对终端中所有应用程序使用资源的调度，管理应用程序的使用状态。任何具有有限资源的系统都需要某种方式管理这些有限的资源，并且保证任何资源的争用的公平的分配和应用程序的稳定性。所有的应用程序都需要注册资源的使用状态，如果资源管理器发现多个应用程序有资源使用的冲突，它将通过既定的规则进行资源调度。详细的功能包括：

1. 资源使用状态注册；
2. 资源使用分配调度；
3. 资源释放管理等。

6.1.10 应用管理

应用管理负责对各种应用程序的运行进行协调管理，并管理应用的下载和升级。

6.1.11 插件管理

插件管理负责对各种加载到终端的插件进行管理，包括插件的下载、注册、升级、调用及注销等。

6.1.12 通信管理

通信管理负责终端中进程之间的通信方式、通信机制的管理，各插件和中间件模块之间信息的传递和调用关系的管理。

6.1.13 API管理

API管理负责中间件业务功能模块的管理及各业务功能模块内部和插件内部的API的管理，应用对API的查询等。

6.1.14 中间件注册管理

中间件注册管理负责中间件业务功能模块的认证注册管理，升级和注销等生命周期管理。

6.1.15 图形用户界面

图形用户界面负责向应用层提供各种图形元素的实现，如按钮、窗口、线条等的实现和管理。

6.1.16 呈现引擎

呈现引擎负责对应用和业务进行解析并呈现给用户。

6.2 资源抽象层

资源抽象层包括系统抽象层接口和硬件抽象层接口。资源抽象层的目的是将具体的硬件平台实现与中间件内部模块设计隔离开来，使得同一套中间件软件能够平滑的移植到不同的硬件平台。

该层仅为硬件平台和中间件软件内部提供匹配接口，不同的中间件软件可以有不同的资源抽象层实现。资源抽象层被用来与第三方机顶盒集成。也就是说对于任何第三方机顶盒厂家，只要实现了本层的软件，客户端中间件和应用程序都可以运行在任何机顶盒厂家的机顶盒里，不需要对不同的硬件和操作系统做适配。

6.2.1 系统抽象层接口

系统抽象层（SAL）负责对操作系统进行抽象，它封装了所有的通用操作，包括任务、定时器、消息队列等，以及一些公共的数据结构，如队列、堆栈等。

系统抽象层屏蔽了操作系统，因此它应该能支持多种操作系统，如VxWorks、Linux、Windows CE等等。通过使用SAL层接口，应用程序和系统软件层不需要关心当前使用的是哪种操作，而且这些应用软件能非常容易地移植到不同地操作系统平台。

系统抽象层应该至少支持下列功能：

1. 进程管理；
2. 线程管理；
3. 信号量的管理；
4. 消息队列功能，包括创建、删除、读、写等；
5. 内存管理；
6. 定时器管理；
7. 网络操作；
8. 日期时间函数支持等。

6.2.2 硬件抽象层接口

硬件抽象层（HAL）负责对底层硬件平台进行抽象，它封装了所有底层硬件平台和接口设备的操作和接口，包括Flash、RAM、键盘、鼠标、遥控器等。接口抽象层维护一个来自于各种输入设备的键值队列，并且负责把这些键值传送到应用层。通过这种方式，应用层不需要关心当前使用的是哪种输入设备，而仅仅需要知道当前的键值以及怎么处理这些键值。为了达到这个目的，一个为所有输入设备服务的统一的键值映射表应该被定义。

这层完美地屏蔽了所有硬件平台信息，主要功能包括：

- 1. 配置信息读写接口；
- 2. 状态灯操作接口；
- 3. 电源控制接口；
- 4. 文件读写接口；
- 5. CPU 管理接口；
- 6. 音视频接口；
- 7. 内存管理接口；
- 8. 遥控器管理接口等。

7 设备驱动及资源系统

设备驱动作为实现IPTV数据接收、解码、数据处理、显示等业务的基本硬件平台接口，是中间件软件平台调用的主要模块。

设备驱动提供给上层应用或中间件软件平台各种各样的功能，从某种角度来说这些功能是一种系统资源，因此设备驱动模块又可称为资源模块。驱动模块或资源模块的上层接口为中间件软件平台移植接口，它包括实时操作系统（RTOS）、图形、消息管理、内存管理和设备驱动等几大子系统。

7.1 设备驱动

设备驱动组成如图4所示。

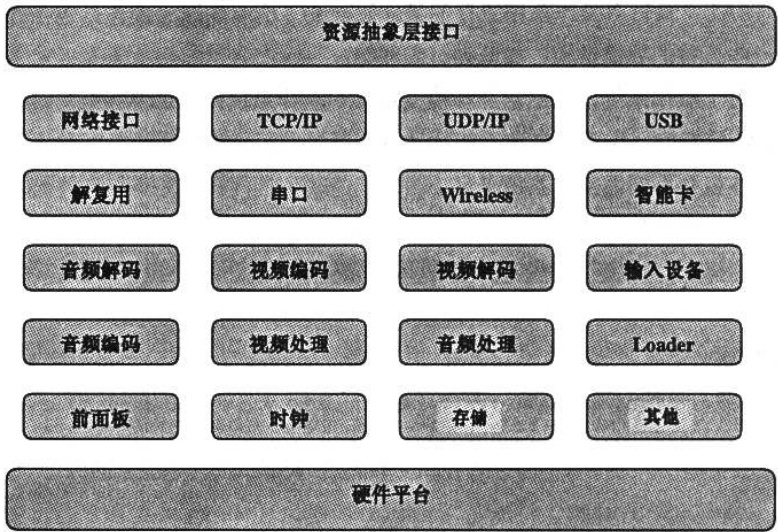


图 4 设备驱动组成

主要资源模块有：

1. 宽带网络接口模块XDSL Modem, Ethernet;
2. 串口模块Serial Port;
3. 解复用模块Demux;
4. TCP/IP模块;
5. UDP/IP模块;
6. 智能卡模块SmartCard;
7. 音频解码模块Audio Decoder;
8. 视频解码模块Video Decoder;
9. 视频编码模块Video Encoder;
10. 音频编码模块Audio Encoder;
11. 音频处理模块;
12. 视频处理与显示控制;
13. 输入设备模块（遥控器、前面板按键等）Input Device;
14. 前面板显示模块Front Panel;
15. Loader;
16. 存储模块Flash/EEPROM/HDD。

7.2 资源层软件架构

资源模块层可以有不同的结构，本规范不对资源模块层的结构设计做出具体的硬性规定，各个软件设计可根据不同的硬件平台及驱动层模块的具体功能做出不同的设计方案。

不管方案如何，各个模块之间应有一定的独立性。这意味着硬件平台所提供的各个设备资源模块均被看作一个个独立的模块，对这些模块的调用均可按对象方法看待，任何调用均看作对资源的利用。对于某些临界保护的模块，其对象是一个实际的驱动，执行驱动对应的硬件操作；而某些对象是一个虚拟的设备，在调用该对象时由设备服务器层进行处理和解释，并执行相应的功能，但没有实际唯一对应的设备进行响应，它或许是几个设备的一个组合，或许是一个逻辑的数据处理。

资源模块具有下面一些特征：

1. 每个资源模块均对应着一个或几个相应的底层驱动设备，此设备由硬件或软件组成。
2. 每个资源模块相对于其他的资源模块而言单独运行。它控制自身的行为，处理中断，并通过事件/消息和其他资源模块或应用程序间单独通信。
3. 中间件软件平台可以在多个线程里同时或不同时地调用同一个资源模块功能。对相同资源模块的调用及其环境值的保护均由资源模块封装层进行协调和管理。
4. 资源驱动模块的功能调用可以是实际对应的硬件操作，如解调，也可以是虚拟的设备，如数据装载等。
5. 对于应用层，每一个资源模块的调用都看作是一独立的行为，各资源模块之间的相互关系和系统资源共享等处理是系统设计和开发者的任务。
6. 应用程序对资源模块请求数据的回传方式有两种，一种是同步返回，即直接通过调用接口返回值返回；一种是异步返回，即驱动层执行到需要回传数据给客户端时以消息方式传给应用程序。

7.3 运行和通信机制

应用线程对设备驱动模块的每次调用可能是同步或异步的。同步调用为阻塞方式，被调函数返回的请求信息是可用的；异步调用为非阻塞方式，在被调函数返回时，请求信息不可用或部分不可用，设备驱动或资源模块发送一条消息给调用者，以返回有效数据和执行状态。

运行机制是对应用线程而言的，而运行策略可能不同。

7.3.1 通信机制

在异步调用过程中，资源驱动模块层的异步信息或事件信息通过事件(Event)发送给应用线程，发给应用程序的事件统一通过一个事件队列进行处理。每个事件由一个类型和代码 (Type, Code) 唯一确定，调用的结果通过事件中的数据指针 (Data) 返回给事件处理者。基于事件原理的通信机制也可以在其他情况下使用，如一个线程向其他线程发送信息。

在使用事件通信机制时，目标线程必须首先向事件管理器注册，给出需要监听的事件类型和它的回调函数地址。当事件到来时，事件管理器调用回调函数并传递给它一个事件指针，并且将其存储的客户数据(Client Data)返回给回调函数。

在多个线程可以执行的情况下，由实时操作系统调度程序负责决定执行哪一个线程。

所有的线程（包括驱动层线程和应用程序线程）均可以使用事件管理器，所以每个事件管理函数应当是线程安全的。

8 插件管理

插件程序是一个功能性的集合，这些功能可以被添加到中间件软件平台中，以便对本规范之外的增值应用和内容等提供解释。这一技术的目的是提供对已有交互系统及应用的支持。

插件程序的恰当定制和插件的选择必须由终端用户来决定，这样做是为了能够自主地选择服务来源。该选择可以通过多种方法来实现，包括购买含有内置插件程序的设备、选择一个下载、或者在没有内存限制的情况下自动选择一个下载。

如果终端设计允许的话，插件应用程序可以驻留在内存里。但平台的运行方式不因插件的使用而改变。

8.1 插件的提供方式

应用可以是事先内嵌的，或者以插件的方式下载的

插件程序有两种可能的实现方案，如图5所示。

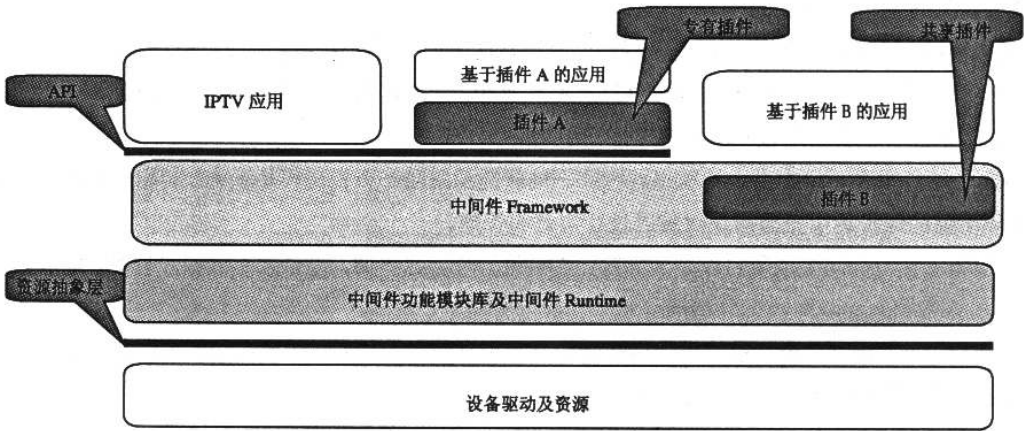


图 5 插件的提供方式

- 和中间件软件平台一起嵌入的 C、C++ 代码或其他本地代码；
- 作为一个应用程序，此应用可以成为系统程序的一部分。

8.2 安全模式

插件程序必须能够充分地利用平台中的资源，以便能和现有平台进行相互操作。一个内置嵌入的插件程序能够利用许多资源而不受终端安全模式的约束。插件程序自己负责管理应用程序执行的安全性。

如果一个插件程序为了提供一个与现有的非插件程序所支持的等同的功能而需要利用一些并不对所有下载的应用都开放的资源，则它需要适当的验证。

9 中间件的发行认证机制

中间件软件是预先开发的，存储在版本服务器上。中间件的发行分两种情况：一种是中间件软件作为机顶盒整体软件包的一部分，随机顶盒整体软件包一起下载安装；另外一种中间件软件业务模块作为独立的插件软件包下载安装。

9.1 中间件的发行注册认证流程

当中间件作为机顶盒整体软件包一部分随整体软件包下载时，它的认证随整体软件包一起完成，不需要单独认证。

当中间件作为独立的插件软件包下载安装时，应该遵循下面的流程：

1. 版本服务器中存储各个版本的插件软件，为机顶盒提供下载。插件软件可由下列参数唯一确定：机顶盒型号、机顶盒需要连接的业务系统的厂商ID、插件的版本编号。
2. 机顶盒启动相应应用时，根据应用需要向版本服务器请求插件的安装。机顶盒向版本服务器提供机顶盒的型号和需要连接的业务系统的厂商ID，并提供机顶盒中当前软件的版本。
3. 版本管理器根据机顶盒提供的信息确定机顶盒需要下载的软件，并向机顶盒提供一个该软件包含的文件的列表。机顶盒根据文件列表向版本服务器请求下载软件包含的各个文件。
4. 版本服务器根据机顶盒的实际情况向机顶盒提供适当的经过签名的中间件软件包。

附录 A
(规范性附录)
中间件软件的数字签名

机顶盒所下载的中间件软件包需要进行数字签名。软件签名机制如图 A.1 所示。

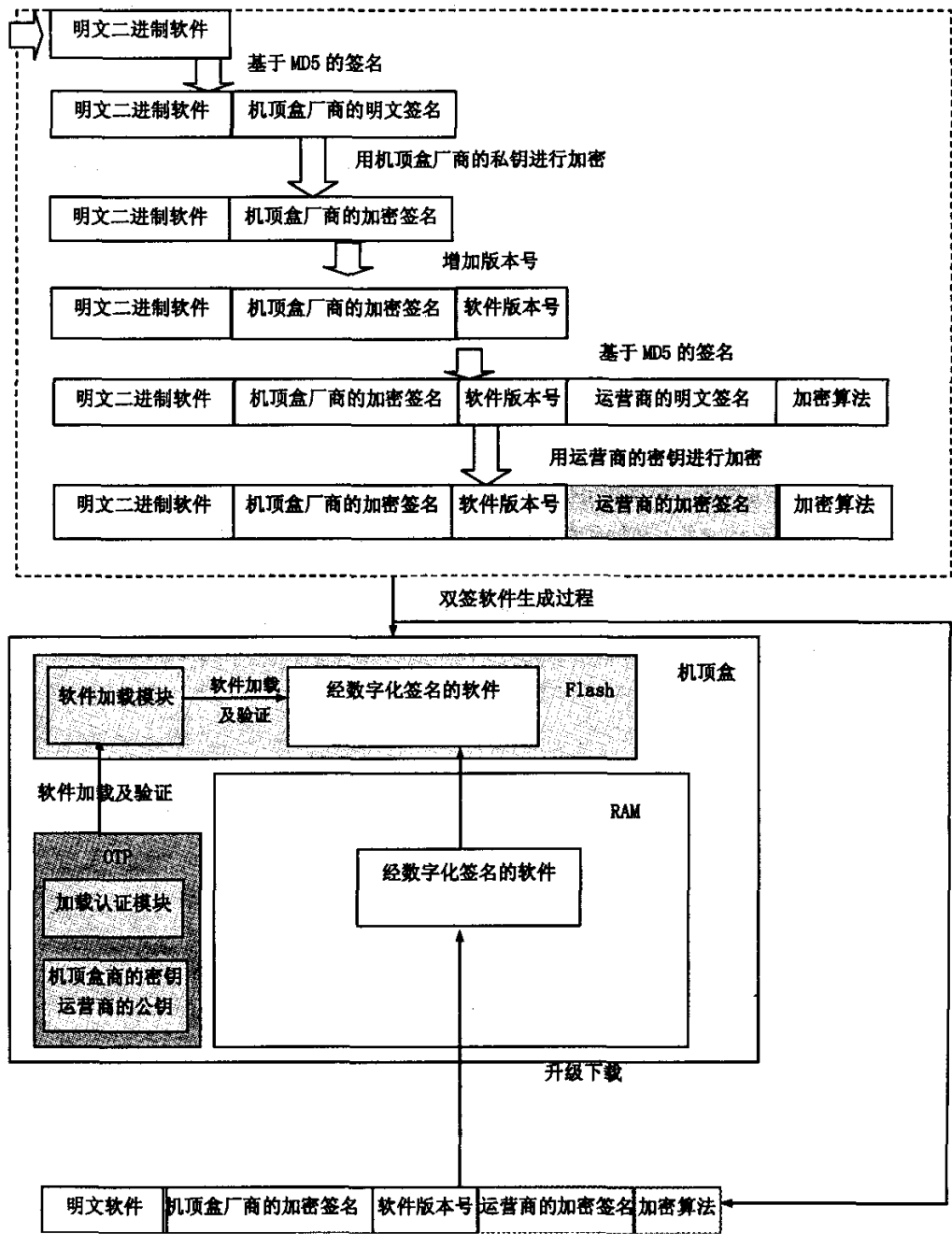


图 A.1 中间件软件包数字签名结构

如果采用运营商和机顶盒厂商双签名的机制，应首先由机顶盒厂商对软件明文文件签名，再由运营商对软件签名。

机顶盒厂家签名部分的校验方法由机顶盒厂商自定。机顶盒厂商也可以不对软件进行签名，但签名的顺序必须符合本节规定。

加密算法标识字段为字符串，取值和含义如下：

- ‘NULL’，表示不带签名
- ‘RSA’，表示 RSA 算法
- ‘3DES’，表示 3DES 算法（ECB 模式）

A.1 加载认证模块的工厂处理流程

1. 机顶盒厂商准备好公私密钥对；
2. 机顶盒厂商公钥和加载认证模块软件在机顶盒出场时烧录到机顶盒的 ROM 中。

A.2 中间件软件模块的发行处理流程

1. 机顶盒厂商准备好二进制明文中间件软件包；
 2. 基于 MD5 的机顶盒厂商明文签名附在二进制明文中间件软件包后面；
 3. 用机顶盒厂商的私钥对签名进行加密，然后替换机顶盒厂商的明文签名；
 4. 对本次的软件生成一个版本号，附加在带有机顶盒厂商签名的软件后面；
 5. 运营商对二进制明文软件、机顶盒厂商的签名、版本号进行 MD5 运算得到运营商的明文签名，并附加它的明文签名及加密算法在带有版本号的软件后面；
 6. 用运营商的密钥对运营商的明文签名进行加密。
-