



# 中华人民共和国国家标准

GB/T 33780.3—2017

---

## 基于云计算的电子政务公共平台技术规范 第3部分：系统和数据接口

Technical specification of electronic government common platform based on cloud computing—Part 3: Interface between system and data

2017-05-31 发布

2017-12-01 实施

中华人民共和国国家质量监督检验检疫总局 发布  
中国国家标准化管理委员会



## 目 次

前言 .....	III
引言 .....	IV
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语、定义和缩略语 .....	1
4 概述 .....	2
5 网络系统互联互通接口 .....	2
6 主机系统接口 .....	3
7 数据库版本转换接口 .....	4
8 数据共享接口 .....	5
9 应用系统与应用服务器中间件接口 .....	5
10 应用服务器中间件互操作接口 .....	6
11 应用系统互操作调用接口 .....	6
12 公共平台资源服务管理接口 .....	6
附录 A (资料性附录) 系统和数据接口关系图 .....	8
附录 B (资料性附录) 接口方法描述 .....	14



## 前 言

GB/T 33780《基于云计算的电子政务公共平台技术规范》预计分为以下 6 个部分：

- 第 1 部分：系统架构；
- 第 2 部分：功能和性能；
- 第 3 部分：系统和数据接口；
- 第 4 部分：操作系统基本要求；
- 第 5 部分：信息资源开放共享系统架构；
- 第 6 部分：服务测试。

本部分为 GB/T 33780 的第 3 部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由中华人民共和国工业和信息化部提出。

本部分由中华人民共和国工业和信息化部(通信)归口。

本部分起草单位：西安未来国际信息股份有限公司、浙江建达科技股份有限公司、湖南科创信息技术股份有限公司、北京中软国际信息技术有限公司、中国信息通信研究院、中国移动通信集团有限公司、曙光信息产业股份有限公司、神州数码系统集成服务有限公司。

本部分主要起草人：王茜、葛新、刘守仓、何舰、邵云华、段延辉、马飞、王凤、张晓辉、郑颖。

## 引 言

电子政务发展正处于转变发展方式、深化应用和突出成效的关键转型期。政府职能转变和服务型政府建设对电子政务发展提出了更新更高要求。以云计算为代表的新兴信息技术、产业、应用不断涌现,深刻改变了电子政务发展技术环境及条件。构建基于云计算的电子政务公共平台可以充分发挥既有资源的作用和新兴信息技术潜能,加快电子政务发展创新,提高应用支撑服务能力,增强安全保障能力,减少重复建设、避免各自为政和信息孤岛。

在电子政务公共平台建设过程中客观形成了物理上分布、功能相互独立的基础设施资源和应用系统等信息系统资源。基于云计算的电子政务公共平台要充分利用这些信息系统资源,通过虚拟化等云计算技术,形成逻辑上统一的“资源池”和面向用户的“服务交付渠道”,在此过程中产生了各种类型应用系统之间的接口关系。本部分主要解决应用系统迁移、分布部署的电子政务公共平台通过互联互通形成更大范围的资源服务池过程中出现的技术接口问题。因此,在电子政务公共平台资源和服务构成要素中,网络、主机、存储、操作系统、数据库管理系统、应用服务器中间件等已经具有的接口技术规范和相互之间已经具有的接口技术规范,在本部分中将加以引用;在与应用系统实施系统集成、部署、迁移过程中产生的接口需求,本部分将提出系统和数据访问接口的接口功能、访问协议、访问方式等技术要求。

本部分不规定接口的具体实现,也不规定接口的应用指南。公共平台服务提供机构可依据本部分提出的系统和数据接口技术规范实施接口实例,公布接口应用方法。

# 基于云计算的电子政务公共平台技术规范 第3部分：系统和数据接口

## 1 范围

GB/T 33780 的本部分规定了电子政务公共平台之间的系统和数据接口的接口功能、访问协议、访问方式等技术要求。

本部分适用于基于云计算的电子政务公共平台。

注：除非特殊说明，以下各章中“电子政务公共平台”和“公共平台”均指“基于云计算的电子政务公共平台”。

## 2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 21062—2007 政务信息资源交换体系

GB/T 21063—2007 政务信息资源目录体系

GB/T 34078.1—2017 基于云计算的电子政务公共平台总体规范 第1部分：术语和定义

## 3 术语、定义和缩略语

### 3.1 术语和定义

GB/T 34078.1—2017 界定的术语和定义适用于本文件。

### 3.2 缩略语

下列缩略语适用于本文件。

API：应用程序编程接口 (Application Programming Interface)

B/S：浏览器/服务器模式 (Browser/Server)

DHCP：动态主机配置协议 (Dynamic Host Configuration Protocol)

DNAT：动态地址转换 (Dynamic Net Address Trancelate)

DNS：域名系统 (Domain Name System)

FTP：文件传输协议 (File Transfer Protocol)

HTTP：超文本传送协议 (Hypertext Transfer Protocol)

IP：网络之间互联的协议 (Internet Protocol)

iSCSI：小型计算机系统接口 (Internet Small Computer System Interface)

JMS：Java 消息服务 (Java Message Service)

JSON：轻量级的数据交换格式 (JavaScript Object Notation)

MAN：城域网 (Metropolitan Area Network)

NAT：网络地址转换 (Network Address Translation)

NFS：网络文件系统 (Network File System)

OTN：光传送网 (Optical Transport Network)

## GB/T 33780.3—2017

PIN:个人识别密码(Personal Identification Number)  
 PPP:点对点协议(Point to Point Protocol)  
 RESTful:一个架构样式的网络系统(Representational State Transfer)  
 SNAT:静态地址转换(StaticNet Address Trancelate)  
 SDH:同步数字体系(Synchronous Digital Hierarchy)  
 SOAP:简单对象访问协议(Simple Object Access Protocol)  
 SONET:同步光纤网络(Synchronous Optical Network)  
 SQL:结构化查询语言(Structured Query Language)  
 TCP:传输控制协议(Transmission Control Protocol)  
 URL:统一资源定位符(Uniform Resource Locator)  
 VLAN:虚拟局域网(Virtual Local Area Network)  
 WSDL:描述 Web 服务和通信的 XML 语言(Web Services Description Language)  
 XML:可扩展标记语言(Extensible Markup Language)

## 4 概述

为满足应用系统集成、部署、迁移和运行的需求,根据公共平台管理对象的划分,电子政务公共平台和应用系统应提供网络系统互联互通接口、主机系统接口、数据库版本转换接口、数据共享接口、应用系统与应用服务器中间件接口、应用服务器互操作接口、应用系统互操作调用接口、公共平台资源服务管理接口。系统和数据接口应封装操作过程实现细节,只向外暴露访问方式和输入参数。

根据电子政务公共平台管理对象之间的关联关系,主机系统接口包括虚拟主机与网络系统接口、虚拟主机与存储设备接口、虚拟主机与应用服务器接口、虚拟主机和应用系统接口。

电子政务公共平台之间的资源管理、调度和监控通过资源服务管理接口完成,公共平台之间不能直接进行互操作。

系统与数据接口关系图参见附录 A。

## 5 网络系统互联互通接口

网络系统互联互通接口的要求如下:

- a) 接口名称:pt1。
- b) 协议:以太网、PPP。
- c) 接口功能:应基于通信运营商或自建传输网所提供的 SDH、OTN、裸光纤等不同传输协议的传输线路,通过局域网络接入标准,骨干传输网接入标准达到互联互通,具体包括:
  - 1) 局域网络接入接口:应支持电口或光口两种,其中电口应支持 100 M 和 1 000 M。100 M 以太网电口,应基于 5 类双绞电缆连接,距离不超过 100 m;1 000 M 以太网电口,应基于 6 类双绞电缆连接,距离不超过 100 m;光口,应基于综合布线室内光缆连接,末端接口应支持 LC、SC、FC 等不同的光纤接口类型。
  - 2) 骨干传输网接入接口:应基于 SDH 传输协议,支持 E1 接口或 CPOS 接口,应提供 2 M/4 M/8 M~155 M 等带宽需求;基于波分协议传输,应支持电口或光口,可提供最低 100 M~1 000 M 等带宽需求。电口接口标准参照局域网络接入接口电口标准,光口接口标准参照局域网络接入接口光口标准;基于裸光纤传输,应支持 LC、SC 接口,可提供大于等于千兆带宽,接口标准参照局域网络接入接口光口标准。
  - 3) 局域网络接入接口和骨干传输网接入接口应适应随着网络带宽不断增加而产生的接口类型



变化。

- d) 接口方法:参见附录 B 中的 B.1。

## 6 主机系统接口

### 6.1 虚拟主机与网络系统接口

虚拟主机与网络系统接口的要求如下:

- a) 接口名称:pt2\_1。
- b) 协议:http/https。
- c) 调用参数:输入{操作上下文,路由名称,路由标识,代理标识,外部网关端口,接口名称,外部网关无类间路由,接口信息,网络标识,主机标识,网络名称,子网,租户标识,端口标识,浮动 IP,防火墙规则,防火墙规则 ID,防火墙策略}。
- d) 返回值:{逻辑值 true/false,网络信息,路由信息,防火墙信息}。
- e) 接口功能:
  - 1) 网络系统应为虚拟主机提供高速存储网络;
  - 2) 网络系统应为虚拟主机提供可靠管理网络;
  - 3) 网络系统应为虚拟主机提供高速虚拟网络;
  - 4) 网络系统应为虚拟主机提供高速公共网络;
  - 5) 虚拟主机应提供基于 L3 层虚拟路由功能;
  - 6) 虚拟主机应提供基于 DHCP 协议的 IP 分配功能;
  - 7) 虚拟主机应提供基于 DNS 协议的域名解析功能;
  - 8) 虚拟主机应提供基于 L2 层虚拟链接交换功能;
  - 9) 虚拟主机应提供基于 iptable 的 NAT 转发功能;
  - 10) 虚拟主机应提供基于 L4-7 层虚拟防火墙策略功能;
  - 11) 虚拟主机应提供基于虚拟设备(虚拟网卡)与虚拟主机之间的映射功能。
- f) 接口方法:参见 B.2.1。

### 6.2 虚拟主机与应用服务器中间件接口

虚拟主机与应用服务器中间件接口的要求如下:

- a) 接口名称:pt2\_2。
- b) 协议:http/https。
- c) 调用参数:输入{主机 IP 或域名,端口,用户名,密码,登录上下文,源文件,目标,校验和}。
- d) 返回值:{逻辑值 true/false,成功/失败}。
- e) 接口功能:
  - 1) 虚拟主机应为应用服务器中间件提供软件包本地上传与下载功能;
  - 2) 虚拟主机应为应用服务器中间件提供软件包校验与防病毒;
  - 3) 虚拟主机应为应用服务器中间件提供软件包在线更新与下载功能;
  - 4) 虚拟主机应为应用服务器中间件提供版本管理功能;
  - 5) 虚拟主机与应用服务器中间件提供编排功能。
- f) 接口方法:参见 B.2.2。

### 6.3 虚拟主机与应用系统接口

虚拟主机与应用系统接口的要求如下:

## GB/T 33780.3—2017

- a) 接口名称:pt2\_3。
- b) 协议:http/https/FTP。
- c) 调用参数:输入{主机 IP 或域名,端口,用户名,密码,登录上下文,源文件,目标,校验和,虚拟主机 ID,目标服务器,资源池名称,资源池描述,提供者,子网,协议,轮询方法}。
- d) 返回值:{逻辑值 true/false,成功/失败}。
- e) 接口功能:
  - 1) 虚拟主机应为应用系统软件包提供本地上传与下载功能;
  - 2) 虚拟主机应为应用系统软件包提供校验与防病毒功能;
  - 3) 虚拟主机应为与应用系统提供整体在线迁移功能;
  - 4) 虚拟主机应为应用系统提供负载均衡功能。
- f) 接口方法:参见 B.2.3。

## 6.4 虚拟主机与存储设备接口

虚拟主机与存储设备接口的要求如下:

- a) 接口名称:pt2\_4。
- b) 协议:http/https。
- c) 调用参数:输入{操作上下文,云硬盘 id,云主机 ID,主机名,挂载点}。
- d) 返回值:{逻辑值 true/false,成功/失败}。
- e) 接口功能:
  - 1) 虚拟主机应提供存储设备的挂载与卸载;
  - 2) 虚拟主机应提供存储设备创建与销毁;
  - 3) 虚拟主机应提供存储设备备份与恢复。
- f) 接口方法:参见 B.2.4。

## 7 数据库版本转换接口

数据库版本转换接口要求如下:

- a) 接口名称:pt3。
- b) 接口协议:SOAP/Webservice。
- c) 调用参数:输入{源数据库类型,源数据库版本,源数据库 IP 地址,源数据库服务端口,源数据库名称,源数据库待迁移数据表列表,源数据库访问用户名,源数据库访问用户的密码,目标数据库类型,目标数据库版本,目标数据库 IP 地址,目标数据库服务端口,目标数据库名称,目标数据库数据迁移表列表,目标数据库访问用户名,目标数据库访问用户的密码,待适配和转换 SQL 语句}。
- d) 返回值:{逻辑值 true/false,适配和转换后的 SQL 语句}。
- e) 接口功能:提供应用系统在异构数据库和相同数据库不同版本之间的数据部署和迁移接口。
  - 1) 应提供在关系数据库管理系统、大数据系统等异构系统之间,实现 API 级的数据迁移和数据转换;
  - 2) 应提供在相同数据库管理系统不同版本之间,实现 SQL 语言的适配和转换;
  - 3) 应提供在异构数据库管理系统之间,实现不同数据类型的适配和转换;
  - 4) 应提供在相同数据库管理系统不同版本之间,实现不同数据类型的适配和转换;
  - 5) 应以 XML/JSON 等格式文件为中间媒介,实现在异构数据库管理系统和相同数据库管理系统不同版本之间的数据转换。

f) 接口方法:参见 B.3。

## 8 数据共享接口

数据共享接口要求如下:

- a) 接口名称:pt4。
- b) 接口协议:JMS/SOAP/Webservice。
- c) 调用参数:输入{目录编号,目录元数据和值,目录名称,目录分类编号,目录内容,数据交换发送方路由编号,待交换的数据,数据交换接收方路由编号,认证信息}。
- d) 返回值:{逻辑值 true/false}。
- e) 接口功能:应提供数据资源目录服务接口、数据交换接口、数据转换接口和数据服务接口,实现不同应用系统之间的数据共享。
  - 1) 目录服务接口应提供数据资源的发现、注册、编目、发布和检索的功能,接口技术要求参考 GB/T 21062—2007。
  - 2) 数据交换接口应提供结构化、半结构化和非结构化数据的交换功能,接口技术要求参考 GB/T 21063—2007。
  - 3) 数据转换接口应提供将非标准格式数据转换为标准格式数据的功能。
  - 4) 数据服务接口应提供将数据资源封装为数据服务,并实现数据服务的调度和管理的功能。
- f) 接口方法:参见 B.4。

## 9 应用系统与应用服务器中间件接口

应用系统与应用服务器中间件接口要求如下:

- a) 接口名称:pt5。
- b) 接口协议:http/Soap/RESTful。
- c) 调用参数:输入{源系统描述数据体,目标系统描述数据体,转换等级,系统的目标部署环境描述数据体,目标部署系统描述数据体,部署参数数据体,需要进行迁移的系统描述数据体,需要转换的目标部署系统描述数据体,迁移参数数据体==需要进行监控的系统描述数据体,业务系统监控管理适配器,监控、管理参数数据体,源业务系统服务描述数据体,目标业务系统服务描述数据体,接口转换参数描述数据体}。
- d) 返回值:{转换结果,部署结果,迁移结果,适配结果,服务接口转换结果}。
- e) 接口功能:应提供应用系统与不同的应用服务器以及相同的应用服务器不同版本之间的打包、部署、迁移和服务接口。
  - 1) 打包接口应提供应用系统基于不同应用服务器中间件及相同应用服务器中间件不同版本之间的打包和转换功能;
  - 2) 部署接口应提供应用系统基于不同应用服务器中间件及相同应用服务器中间件不同版本之间的部署功能;
  - 3) 迁移接口应提供应用系统基于不同应用服务器中间件及相同应用服务器中间件不同版本之间的迁移功能;
  - 4) 服务接口应提供应用系统基于不同应用服务器中间件及相同应用服务器中间件不同版本之间运行的管理、监控等服务功能;
  - 5) 应为应用系统提供跨运行平台、跨开发语言的服务调用接口。
- f) 接口方法:参见 B.5。

## 10 应用服务器中间件互操作接口

应用服务器中间件互操作接口要求如下：

- a) 接口名称:pt6。
- b) 协议:http/https/Soap/RESTful。
- c) 调用参数:无。
- d) 返回值:无。
- e) 接口功能:应提供异构应用服务器中间件之间的互操作与应用集群管理。
  - 1) 提供同构应用服务器中间件之间的互操作功能,跨异构应用服务器中间件的互操作具备运行平台无关性和编程语言无关性;
  - 2) 提供应用系统集群接口,同一应用系统的不同实例可跨同构应用服务器中间件实现集群部署;
  - 3) 提供分布式部署接口,应用系统的不同构件可分布部署在异构的应用服务器中间件中。
- f) 接口方法:参见 B.6。

## 11 应用系统互操作调用接口

应用系统互操作调用接口如下：

- a) 接口名称:pt7。
- b) 接口协议:http/Soap/RESTful/Socket。
- c) 调用参数:输入{远程网络地址,远程服务端口,通信内容,本地监听端口,用户唯一标识符,用户口令,数字证书唯一识别码,数字证书 PIN 码}。
- d) 返回值:{通信完成状态标识,身份认证结果标识}。
- e) 接口功能:应提供应用系统与不同的中间件以及相同的中间件不同版本之间的打包、部署、迁移和服务接口。
  - 1) 应提供应用系统之间互调用的方法和参数,支持跨平台、跨编程语言的访问方式;
  - 2) 应支持同步、异步消息通讯的调用方式;
  - 3) 应支持应用系统之间互调用的身份认证和单点登录。
- f) 接口方法:参见 B.7。

## 12 公共平台资源服务管理接口

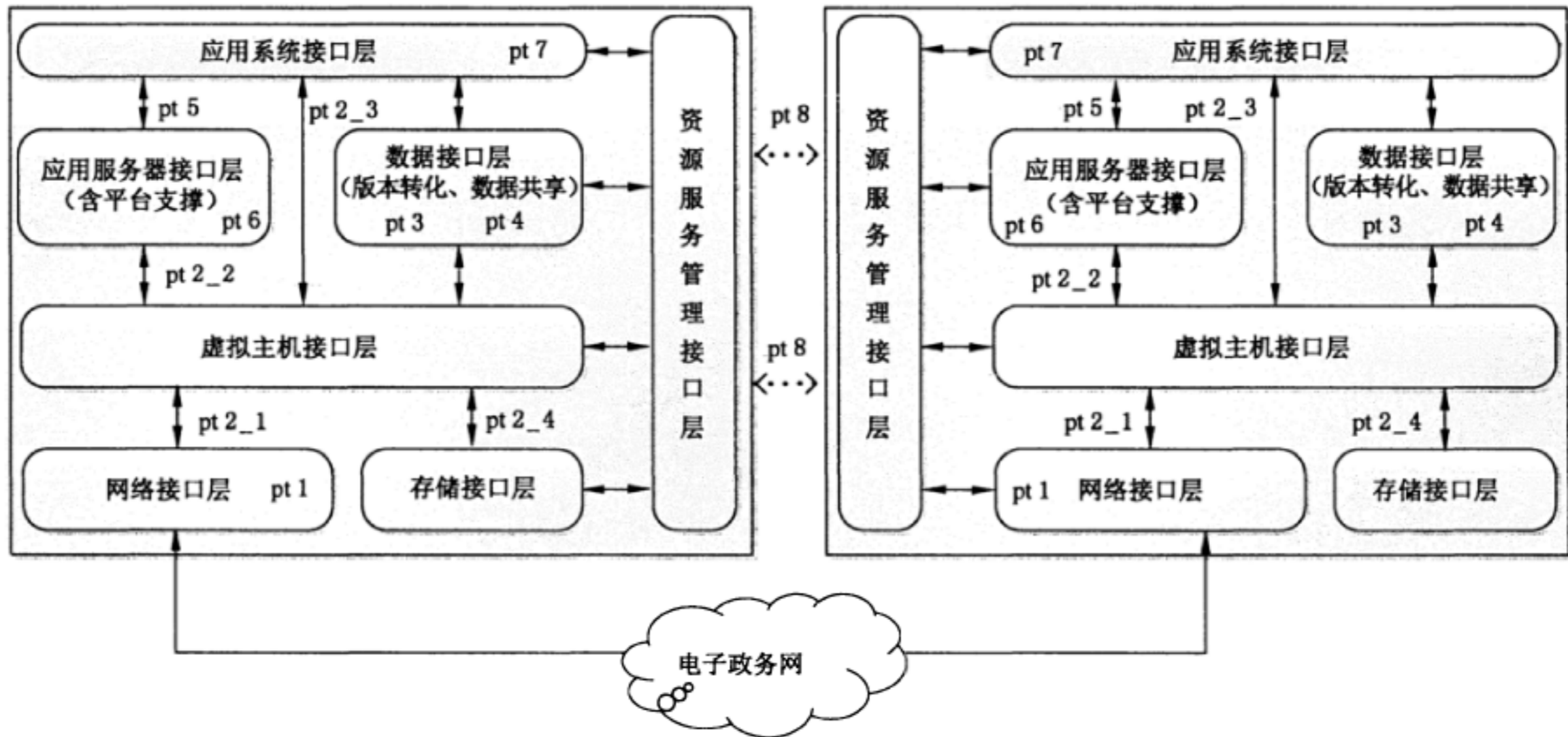
公共平台资源服务管理接口要求如下：

- a) 接口名称:pt8。
- b) 协议:http/https。
- c) 调用参数:输入{用户唯一识别身份编码,用户身份口令,数字证书识别码,资源域编码,模板 id,资源名称,内存和 CPU,资源个数,服务时间,资源 id,平台服务 ip,系统报文}。
- d) 返回值:{认证成功或失败,用户授权可访问信息,资源 id 和名称}。
- e) 接口功能:公共平台应提供跨区域的主机、存储、支撑软件、运维监控、安全保障等服务资源共享与调度访问接口。
  - 1) 应提供远程租户与用户访问的认证与授权功能,通过提供身份认证 key,验证该调用方身份和权限,并获用户账号下可用的服务资源授权清单;

- 2) 应提供虚拟主机资源远程管理与访问功能；
  - 3) 应提供对存储远程管理与操作功能；
  - 4) 应提供支撑软件远程管理功能；
  - 5) 应提供对应用系统的管理与监控功能；
  - 6) 应提供监控信息项跨区域推送功能；
  - 7) 应提供安全支撑服务远程管理功能；
  - 8) 应提供公共平台计算资源地址、数据库资源地址、数据共享资源地址、中间件资源地址的功能。
- f) 接口方法：参见 B.8。

**附录 A**  
**(资料性附录)**  
**系统和数据接口关系图**

**A.1 公共平台系统和数据接口关系**



**图 A.1 公共平台系统和数据接口关系图**

如图 A.1 所示,公共平台之间的接口是资源服务管理接口层(公共平台资源服务管理接口)。公共平台内的资源和服务共享接口模型从底层到上层应用服务依次是网络接口层(网络系统互联互通接口),存储接口层和虚拟主机接口层(主机系统接口)、应用服务器接口层(应用系统与应用服务器接口、应用服务器互操作接口)和数据接口层(数据库版本转换接口、数据共享接口)以及应用系统接口层(应用系统互操作接口);每个接口层的接口按照业务系统部署、迁移和资源服务共享需求定义接口功能、访问协议、访问方法,并遵循国家已发布的相关接口规范。

**A.2 主机系统功能接口对照表**

**A.2.1 虚拟主机与网络功能接口的对照关系见表 A.1~表 A.5。**

**表 A.1 基于 L3 层虚拟路由方法**

序	方法原型	功能描述
1	router_added (router_id, router)	创建一个新的本地路由
2	router_deleted (context,router_id)	删除路由方法
3	add_router_to_l3_agent(context, id, router_id)	添加路由至 L3 代理
4	remove_router_from_l3_agent(context,l3_agent,router_id)	从 L3 代理中删除路由

表 A.1 (续)

序	方法原型	功能描述
5	external_gateway_added (ri, ex_gw_port, interface_name, internal_cidrs)	设置路由的外部网关
6	external_gateway_removed(ri, ex_gw_port, interface_name, internal_cidrs)	从路由器中删除一个外部网关
7	add_router_interface (context, router_id, interface_info)	为路由器添加一个内部网关
8	remove_router_interface(context, router_id, interface_info)	从路由器内移除一个内部网关

表 A.2 基于 L3 层虚拟路由方法

序	方法原型	功能描述
1	list_dhcp_agents_hosting_network(context, network_id)	列出所选网络的 DHCP 代理
2	network_added_to_agent(context, network_id, host)	为网络添加 DHCP 代理
3	network_removed_from_agent(context, network_id, host)	为网络删除 DHCP 代理
4	list_networks_on_dhcp_agent(context, id)	列出拥有 DHCP 代理的所有网络

表 A.3 基于 L2 层虚拟链接交换方法

序	方法原型	功能描述
1	create_network(context, network)	为租户创建网络
2	delete_network(context, id)	删除网络
3	get_networks (context, filters = None, fields = None, sorts = None, limit = None, marker = None, page_reverse = False)	列出网络
4	get_network(context, id, fields = None)	显示网络详细信息
5	update_network(context, id, network)	更新网络信息
6	create_subnet(context, subnet)	为租户创建一个子网
7	delete_subnet(context, id)	删除一个指定的子网
8	get_subnets(context, filters = None, fields = None, sorts = None, limit = None, marker = None, page_reverse = False)	列出所给租户的全部子网
9	get_subnet(context, id, fields = None)	显示指定子网的详细信息
10	update_subnet(context, id, subnet)	更新子网的详细信息
11	port_delete(context, port_id)	删除指定端口
12	get_ports(context, filters = None, fields = None, sorts = None, limit = None, marker = None, page_reverse = False)	列出指定租户的全部端口
13	get_port(context, id, fields = None)	显示指定端口的详细信息
14	update_port(context, id, port)	更新端口的信息
15	set_port_status(session, port_id, status)	强制更改端口状态

表 A.4 基于 iptable 的 NAT 转发方法

序	方法原型	功能描述
1	create_floatingip(context, floatingip)	为指定租户创建浮动 IP
2	delete_floatingip(context, id)	为指定租户删除浮动 IP
3	get_floatingips(context, filters=None, fields=None, sorts=None, limit=None, marker=None, page_reverse=False)	列出所有可用浮动 IP
4	get_floatingip(context, id, fields=None)	显示浮动 IP 的详细信息
5	update_floatingip(context, id, floatingip)	更新浮动 IP 的信息

表 A.5 基于 L4-7 层虚拟防火墙策略方法

序	方法原型	功能描述
1	create_firewall_rule(context, firewall_rule)	创建一个防火墙规则
2	delete_firewall_rule(context, id)	删除一个防火墙规则
3	get_firewall_rules(context, filters=None, fields=None)	列出所有防火墙规则
4	get_firewall_rule(context, id, fields=None)	显示防火墙规则的详细信息
5	update_firewall_rule(context, id, firewall_rule)	更新防火墙规则
6	create_firewall_policy(context, firewall_policy)	创建一个防火墙策略
7	delete_firewall_policy(context, id)	删除一个指定的防火墙策略
8	insert_rule(context, id, rule_info)	为防火墙策略添加一个防火墙规则
9	get_firewall_policies(context, filters=None, fields=None)	列出租户所有的全部防火墙策略
10	remove_rule(context, id, rule_info)	为指定防火墙策略删除一条防火墙规则
11	get_firewall_policy(context, id, fields=None)	显示所给出的防火墙策略的详细信息
12	update_firewall_policy(context, id, firewall_policy)	更新防火墙策略

A.2.2 虚拟主机与应用服务器中间件接口对照关系见表 A.6。

表 A.6 虚拟主机与应用服务器中间件接口功能对照表

序	方法原型	功能描述
1	webssh2(host, port, username, password)	远程登录虚拟主机
2	upload(context, sourcefile, targetdiectory)	上传文件到虚拟机指定目录
3	download(context, sourcefile)	从远程虚拟主机上下载文件到本地
4	checksum(context, sourcefile, checksumcode)	完成软件包校验
5	security_secribe(context, sourcefile)	为上传的软件包实现防病毒扫描
6	install_online(host, package)	为虚拟主机提供在线包更新与安装功能

A.2.3 虚拟主机与应用系统接口对照关系见表 A.7。



表 A.7 虚拟主机与应用系统接口功能对照表

序	方法原型	功能描述
1	webssh2 (host, port, username, password)	远程登录虚拟主机
2	upload (context, sourcefile, targetdiectory)	上传文件到虚拟机指定目录
3	download (context, sourcefile)	从远程虚拟主机上下载文件到本地
4	checksum (context, sourcefile, checksumcode)	虚拟主机比较 checksumcode, 完成软件包校验
5	security_secribe (context, sourcefile)	为上传的软件包实现防病毒扫描
6	migrate_online (context, uuid, dest_host)	为虚拟主机以及虚拟主机上的应用提供在线迁移功能
7	resource_pool(name, description, provide, subnet, protols, method)	增加一个负载均衡资源池
8	addHost (name, member, weight, port)	为资源池增加一个虚拟主机

A.2.4 虚拟主机与存储设备接口对照关系表 A.8。

表 A.8 虚拟主机与存储设备接口功能对照表

序	方法原型	功能描述
1	attach_volume(self, context, volume_id, instance_uuid, host_name, mountpoint, mode)	挂载云硬盘
2	detach_volume(context, volume_id)	卸载云硬盘
3	create(self, req, body)	创建一个云硬盘
4	delete(self, req, id)	删除一个云硬盘
5	update(self, req, id, body)	更新一个云硬盘
6	index(self, req)	列出云硬盘总表
7	detail(self, req)	列出一个云硬盘列表详情
8	create(self, req, body)	创建一个新备份
9	delete(self, req, id)	删除一个备份
10	restore(self, req, id, body)	恢复一个存在的备份云硬盘

A.3 数据库版本转换功能接口对照关系

数据库版本转换功能接口对照关系见表 A.9。

表 A.9 数据库版本转换功能接口对照表

序	方法原型	功能描述
1	data_migrate_sdb_to_tdb(sdb_name, sdb_version, sdb_ip, sdb_port, sdb_dbname, sdb_tablelist, sdb_username, sdb_pwd, tdb_name, tdb_version, tdb_ip, tdb_port, tdb_dbname, tdb_tablelist, tdb_username, tdb_pwd)	实现相同数据库管理系统不同版本之间的数据迁移和转换, 以及不同数据库管理系统之间的数据迁移和转换

表 A.9 (续)

序	方法原型	功能描述
2	data_sql_adapter_sdb_to_tdb(sdb_name, sdb_version, ssql, tdb_name, tdb_version)	实现相同数据库管理系统不同版本之间的 SQL 适配和转换, 以及不同数据库管理系统之间的 SQL 适配和转换

A.4 数据共享接口对照关系

数据共享接口对照关系见表 A.10。

表 A.10 数据共享接口对照表

序	方法原型	功能描述
1	directory_register(token, metadata)	实现目录元数据的采集和注册
2	directory_catalog(token, id, directory_type_id)	将注册的元数据审核后并编目到某一目录分类中
3	directory_publish(token, id)	实现将某一个编目的目录进行发布, 使外部用户可以进行访问
4	directory_retrieval(token, id, directory_name, directory_type_id, directory_content)	根据目录编号, 目录分类编号, 目录名称、目录内容检索目录信息, 目录名称、目录内容检索支持模糊查询, 并返回检索到的目录列表。除用户认证 token 外, 其他参数任选其一
5	data_exchange(token, srouter_id, data, trouter_id_list)	实现数据的交换, 支持结构化和非结构化数据的交换, 支持一对一、一对多

A.5 应用服务与应用服务器中间件接口对照关系

应用服务与应用服务器中间件接口对照关系参见表 A.11。

表 A.11 应用服务与应用服务器中间件接口功能对照表

序	方法原型	功能描述
1	export_app(source_target, dest_target, level)	提供应用系统之间同步数据发送
2	deploy_app(target, app, deploy_content)	提供应用在不同应用服务器中部署的方法
3	migration_app(source_target, dest_target, deploy_content)	提供应用系统之间同步数据发送
4	monitor_app(target_app, app_adapter, monitor_content)	提供应用系统在不同中间件环境中管理, 监控适配的方法
5	service_brtidge(souce_service, target_service, transform_content)	提供应用系统在不同平台, 不同开发语言之间进行服务调用方法

## A.6 应用系统互操作接口对照关系

应用系统互操作接口对照关系参见表 A.12。

表 A.12 应用系统互操作接口功能对照表

序	方法原型	功能描述
1	send_message_sync(remote_ip,remote_port,content)	提供应用系统之间同步数据发送
2	receive_message_sync(local_port, content)	提供应用系统之间同步接收数据
3	send_message_async(remote_ip,remote_port,content)	提供应用系统之间异步数据发送
4	receive_message_async (local_port, content)	提供应用系统之间异步接收数据
5	authenticate_user(user_id,passwd)	用户使用用户名,口令进行用户身份认证
6	authenticate_cert(cert_id,cert_pin)	用户数字证书进行用户身份认证

## A.7 公共平台资源服务管理接口对照关系

公共平台资源服务管理接口对照关系参见表 A.13。

表 A.13 公共平台资源服务管理接口功能对照表

序	方法原型	功能描述
1	authenticate_user(user_id,passwd)	用户使用用户名,口令进行用户身份认证
2	authenticate_cert(cert_id,cert_pin)	用户数字证书进行用户身份认证
3	resources_authorize (user_id,zone_id)	获取用户在一个域内的授权可访问资源的信息
4	getComputerPlatform()	该方法获取计算资源平台地址
5	getDBPlatform()	该方法获取数据库资源平台地址
6	getStoragePlatform()	该方法获取数据共享资源平台地址
7	get MiddlewarePlatform()	该方法获取中间件资源平台地址
8	sendAppMonitorMessage(IPAddress,monitorMessage)	该方法负责将应用系统监控数据推送到平台
9	receiveDeclare()	该方法用于接收各应用系统发送的监控指标的声明报文
10	receiveMonitor()	该方法用于接收各应用系统发送的监控数据

**附 录 B**  
(资料性附录)  
接口方法描述

### B.1 网络系统互连互通接口

网络系统互连互通接口的详细要求如下：

- a) 局域网终端接入。局域网终端用户接口方法,可使用千兆电口直接接入终端,例如:PC、网络打印机、网络传真等。
- b) 局域网内设备间接入。局域网内设备间互联,应根据两端设备提供的接口介质进行接入,例如:一端设备提供 LC 光接口,另一端设备提供 SC 光接口,两端设备连接需要采用 LC-SC 的光纤进行连接;若两端设备都提供千兆电口,两端需要使用标准的 6 类双绞线(网线)进行互联。
- c) 骨干传输网互联接入。骨干传输网互联,应根据两端设备提供的接口介质进行接入,例如:一端设备提供 E1 接口,另一端设备提供 CPOS 业务光接口(接口类型 LC),两端设备连接需要运营商采用 SDH 传输协议提供线路,CPOS 接口要提供 LC 光纤接入;若两端设备都只能提供电口接入,运营商在接入末端需要增加直接提供电口的传输设备,利用双绞线直接接入网络设备;若两端设备可提供光口接入,运营商在接入末端需要提供可直接接入网络设备的光接口类型接入,例如直接提供 LC、SC 接口类型的光纤接入网络设备。

### B.2 主机系统接口

#### B.2.1 虚拟主机与网络系统接口

虚拟主机与网络系统接口实现的示例如下：

- a) 基于 L3 层虚拟路由方法：
  - 1) 创建路由方法：
 

【方法原型】router\_added(router\_id, router)。

【方法功能】创建一个新的本地路由。

【参数说明】router\_id:路由 ID;  
route:路由名称。

【返回值】成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,创建路由。

【提示】创建成功/失败。
  - 2) 删除路由方法：
 

【方法原型】router\_deleted(context,router\_id)。

【方法功能】删除路由方法。

【参数说明】context:操作上下文;  
router\_id:路由 ID。

【返回值】成功/失败。

- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,删除路由。
- 【提示】**删除成功/失败。
- 3) 添加路由到代理方法:
- 【方法原型】**add\_router\_to\_l3\_agent(context, id, router\_id)。
- 【方法功能】**添加路由至 L3 代理。
- 【参数说明】**context:操作上下文;  
id:代理 ID;  
router\_id:路由 ID。
- 【返回值】**成功/失败。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,返回添加路由结果。
- 【提示】**添加成功/失败。
- 4) 从代理中删除路由方法:
- 【方法原型】**remove\_router\_from\_l3\_agent (context, l3\_agent, router\_id)。
- 【方法功能】**从 L3 代理中删除路由。
- 【参数说明】**context:操作上下文;  
l3\_agent:代理 ID;  
router\_id:路由 ID。
- 【返回值】**成功/失败。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,删除代理中的路由。
- 【提示】**删除成功/失败。
- 5) 设置外部网关方法:
- 【方法原型】**external\_gateway\_added (ri , ex\_gw\_port, interface\_name, internal\_cidrs)。
- 【方法功能】**设置路由的外部网关。
- 【参数说明】**ri:路由信息;  
Ex\_gw\_port:外部网关端口;  
Interface\_name:接口名称;  
Internal\_cidrs:外部网关无类间路由。
- 【返回值】**成功/失败。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,设置外部网关。
- 【提示】**设置外部网关成功/失败。
- 6) 删除外部网关方法:
- 【方法原型】**external\_gateway\_removed (ri , ex\_gw\_port,interface\_name, internal\_cidrs)。
- 【方法功能】**从路由器中删除一个外部网关。
- 【参数说明】**ri:路由信息;  
Ex\_gw\_port:外部网关端口;  
Interface\_name:接口名称;  
Internal\_cidrs:外部网关无类间路由。
- 【返回值】**成功/失败。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,删除外部网关。
- 【提示】**外部网关删除成功/失败。
- 7) 添加路由接口方法:
- 【方法原型】**add\_router\_interface (context, router\_id, interface\_info)。
- 【方法功能】**为路由器添加一个内部网关。

- 【参数说明】**context:操作上下文;  
Route\_id:路由 ID;  
Interface\_info:接口信息。
- 【返回值】**router\_iface\_info:路由接口信息。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,添加路由接口信息。
- 【提示】**接口添加/成功。
- 8) 移除路由接口方法:
- 【方法原型】**remove\_router\_interface(context, router\_id, interface\_info)。
- 【方法功能】**从路由器内移除一个内部网关。
- 【参数说明】**context:操作上下文;  
Route\_id:路由 ID;  
Interface\_info:接口信息。
- 【返回值】**del\_ret:删除结果。
- 【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,删除路由接口。
- 【提示】**删除接口成功/失败。
- b) 基于 DHCP 协议的 IP 分配方法:
- 1) 列出所有网络 DHCP 代理方法:
- 【方法原型】**list\_dhcp\_agents\_hosting\_network(context, network\_id)。
- 【方法功能】**列出所选网络的 DHCP 代理。
- 【参数说明】**context:操作上下文;  
network\_id:网络 id。
- 【返回值】**包含列出代理的字典。
- 【使用方法】**发起者调用该接口,传递参数,将会返回包含列出代理的字典。
- 【提示】**无。
- 2) 为网络添加代理方法:
- 【方法原型】**network\_added\_to\_agent(context, network\_id, host)。
- 【方法功能】**为网络添加 DHCP 代理。
- 【参数说明】**context:操作上下文;  
network\_id:网络 id;  
host:主机。
- 【返回值】**成功/失败。
- 【使用方法】**发起者调用该接口,传递参数,将为网络添加指定代理。
- 【提示】**添加代理成功/失败。
- 3) 为网络删除代理方法:
- 【方法原型】**network\_removed\_from\_agent(context, network\_id, host)。
- 【方法功能】**为网络删除 DHCP 代理。
- 【参数说明】**context:上下文;  
network\_id:网络 id;  
host:主机。
- 【返回值】**成功/失败。
- 【使用方法】**发起者调用该接口,传递参数,将删除网络中的指定代理。
- 【提示】**删除代理成功/失败。
- 4) 列出 DHCP 代理所有网络方法:

**【方法原型】**list\_networks\_on\_dhcp\_agent(context,id)。

**【方法功能】**列出拥有 DHCP 代理的所有网络。

**【参数说明】**context:上下文;

id:网络 id。

**【返回值】**包含网络信息的字典。

**【使用方法】**发起者调用该接口,传递参数,将会得到有 DHCP 代理的网络字典。

**【提示】**无。

c) 基于 L2 层虚拟链接交换方法:

1) 创建网络方法:

**【方法原型】**create\_network(context,network)。

**【方法功能】**为租户创建网络。

**【参数说明】**context:上下文;

network:网络名。

**【返回值】**net:网络信息。

**【使用方法】**发起者调用该接口,传递参数,将会得到新创建的网络信息。

**【提示】**创建网络成功/失败。

2) 删除网络方法:

**【方法原型】**delete\_network(context,id)。

**【方法功能】**删除网络。

**【参数说明】**context:上下文;

id:网络 id。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,将会删除该网络。

**【提示】**删除成功/失败。

3) 列出网络方法:

**【方法原型】**:get\_networks(context, filters=None, fields=None, sorts=None, limit=None, marker=None, page\_reverse=False)。

**【方法功能】**列出网络。

**【参数说明】**context:上下文;

filters=None:过滤;

fields=None:字段;

sorts=None:排序;

limit=None:限制;

marker=None:标记;

page\_reverse=False:倒退页面。

**【返回值】**包含所请求网络信息的列表。

**【使用方法】**发起者调用该接口,传递参数,获得满足需求的网络信息列表。

**【提示】**获取成功/失败。

4) 显示网络详情方法:

**【方法原型】**get\_network(context, id, fields=None)。

**【方法功能】**显示网络详细信息。

**【参数说明】**context:上下文;

id:网络 id;

fields=None: 字段。

【返回值】所请求网络详细信息。

【使用方法】发起者调用该接口, 传递参数, 获得该网络的详细信息。

【提示】获取成功/失败。

5) 更新网络方法:

【方法原型】update\_network(context, id, network)。

【方法功能】更新网络信息。

【参数说明】context: 上下文;

id: 网络 id;

network: 网络信息。

【返回值】net: 更新结果。

【使用方法】发起者调用该接口, 传递参数, 执行更新。

【提示】更新成功/失败。

6) 创建子网方法:

【方法原型】create\_subnet(context, subnet)。

【方法功能】为租户创建一个子网。

【参数说明】context: 上下文;

subnet: 子网名称。

【返回值】新创建子网信息。

【使用方法】发起者调用该接口, 传递参数, 创建子网。

【提示】创建成功/失败。

7) 删除子网方法:

【方法原型】delete\_subnet(context, id)。

【方法功能】删除一个指定的子网。

【参数说明】context: 上下文;

id: 子网 id。

【返回值】成功/失败。

【使用方法】发起者调用该接口, 传递参数, 删除该子网。

【提示】删除成功/失败。

8) 列出子网方法:

【方法原型】get\_subnets(context, filters=None, fields=None, sorts=None, limit=None, marker=None, page\_reverse=False)。

【方法功能】列出所给租户的全部子网。

【参数说明】context: 上下文;

filters=None: 过滤;

fields=None: 字段;

sorts=None: 排序;

limit=None: 限制;

marker=None: 标记;

page\_reverse=False: 倒退页面。

【返回值】返回子网信息列表。

【使用方法】发起者调用该接口, 传递参数, 返回子网信息列表。

【提示】获取成功/失败。



- 9) 显示指定子网方法：  
**【方法原型】**get\_subnet(context, id, fields=None)。  
**【方法功能】**显示指定子网的详细信息。  
**【参数说明】**context:上下文；  
                   id:子网 id；  
                   fields=None:字段。  
**【返回值】**返回该子网信息。  
**【使用方法】**发起者调用该接口,传递参数,返回该子网信息。  
**【提示】**获得成功/失败。
- 10) 更新子网方法：  
**【方法原型】**update\_subnet(context, id, subnet)。  
**【方法功能】**更新子网的详细信息。  
**【参数说明】**context:上下文；  
                   id:子网 id；  
                   Subnet:子网信息。  
**【返回值】**成功/失败。  
**【使用方法】**发起者调用该接口,传递参数,更新子网。  
**【提示】**更新成功/失败。
- 11) 创建端口方法：  
**【方法原型】**create\_port。  
**【方法功能】**为指定租户创建端口。  
**【参数说明】**context:上下文；  
                   port:端口名。  
**【返回值】**返回端口信息。  
**【使用方法】**发起者调用该接口,传递参数,返回关口信息。  
**【提示】**创建成功/失败。
- 12) 删除端口方法：  
**【方法原型】**port\_delete(context, port\_id)。  
**【方法功能】**删除指定端口。  
**【参数说明】**context:上下文；  
                   port\_id:端口 id。  
**【返回值】**成功/失败。  
**【使用方法】**发起者调用该接口,传递参数,删除该端口。  
**【提示】**删除成功/失败。
- 13) 获取端口方法：  
**【方法原型】**get\_ports(context, filters=None, fields=None, sorts=None, limit=None, marker=None, page\_reverse=False)。  
**【方法功能】**列出指定租户的全部端口。  
**【参数说明】**context:上下文；  
                   filters=None:过滤；  
                   fields=None:字段；  
                   sorts=None:排序；  
                   limit=None:限制；

marker=None:标记。

【返回值】返回端口列表信息。

【使用方法】发起者调用该接口,传递参数,返回端口列表。

【提示】获得信息成功/失败。

14) 获取指定端口方法:

【方法原型】get\_port(context,id,fields=None)。

【方法功能】显示指定端口的详细信息。

【参数说明】context:上下文;

id:端口 id;

fields=None:字段。

【返回值】返回指定端口信息。

【使用方法】发起者调用该接口,传递参数,返回指定端口信息。

【提示】获取成功/失败。

15) 更新端口方法:

【方法原型】update\_port(context, id, port)。

【方法功能】更新端口的信息。

【参数说明】context:上下文;

id:端口 id;

port:端口名。

【返回值】成功/失败。

【使用方法】发起者调用该接口,传递参数,更新端口。

【提示】更新成功/失败。

16) 设定端口方法:

【方法原型】set\_port\_status(session,port\_id,status)。

【方法功能】强制更改端口状态。

【参数说明】session:会话;

port\_id:端口 id;

status:状态。

【返回值】成功/失败。

【使用方法】发起者调用该接口,传递参数,设置端口状态。

【提示】设置成功/失败。

d) 基于 iptable 的 NAT 转发方法:

1) 创建浮动 IP 方法:

【方法原型】create\_floatingip(context, floatingip)。

【方法功能】为指定租户创建浮动 IP。

【参数说明】context:上下文;

floatingip:浮动 IP 名;

ID:浮动 IP 标识。

【返回值】成功/失败。

【使用方法】发起者调用该接口,传递参数,返回创建结果。

【提示】创建成功/失败。

2) 删除浮动 IP 方法:

【方法原型】delete\_floatingip(context,id)。

**【方法功能】**为指定租户删除浮动 IP。

**【参数说明】**context:上下文;

id:浮动 IP 的 id 号。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,返回删除结果。

**【提示】**删除成功/失败。

3) 获取浮动 IP 列表方法:

**【方法原型】**get\_floatingips(context, filters=None, fields=None, sorts=None, limit=None, marker=None, page\_reverse=False)。

**【方法功能】**列出所有可用浮动 IP。

**【参数说明】**context:上下文;

filters=None:过滤;

fields=None:字段;

sorts=None:排序;

limit=None:限制;

marker=None:标记。

**【返回值】**返回该浮动 IP。

**【使用方法】**发起者调用该接口,传递参数,返回该浮动 IP。

**【提示】**获取成功/失败。

4) 获取浮动 IP 详细信息方法:

**【方法原型】**get\_floatingip(context, id, fields=None)。

**【方法功能】**显示浮动 IP 的详细信息。

**【参数说明】**context:上下文;

id:浮动 IP 的 id 号;

fields=None:字段。

**【返回值】**返回浮动 IP 详细信息。

**【使用方法】**发起者调用该接口,传递参数,返回详细信息。

**【提示】**获取成功/失败。

5) 更新浮动 IP 方法:

**【方法原型】**update\_floatingip(context, id, floatingip)。

**【方法功能】**更新浮动 IP 的信息。

**【参数说明】**context:上下文;

id:浮动 IP 的 id 号;

floatingip:浮动 IP 名。

**【返回值】**返回浮动 IP 列表信息。

**【使用方法】**发起者调用该接口,传递参数,更新浮动 IP。

**【提示】**更新成功/失败。

e) 基于 L4-7 层虚拟防火墙策略方法:

1) 创建防火墙规则方法:

**【方法原型】**create\_firewall\_rule(context, firewall\_rule)。

**【方法功能】**创建一个防火墙规则。

**【参数说明】**context:上下文;

firewall\_rule:防火墙规则。

**【返回值】**返回防火墙规则。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙规则。

**【提示】**创建成功/失败。

2) 删除防火墙规则方法:

**【方法原型】**delete\_firewall\_rule(context, id)。

**【方法功能】**删除一个防火墙规则。

**【参数说明】**context:上下文;  
id:防火墙规则的 id。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,返回删除结果。

**【提示】**删除成功/失败。

3) 列出所有防火墙规则方法:

**【方法原型】**get\_firewall\_rules(context, filters=None, fields=None)。

**【方法功能】**列出所有防火墙规则。

**【参数说明】**context:上下文;  
filters=None:过滤;  
fields=None:字段。

**【返回值】**返回包含防火墙规则的字典。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙规则列表。

**【提示】**获取成功/失败。

4) 显示防火墙规则方法:

**【方法原型】**get\_firewall\_rule(context, id, fields=None)。

**【方法功能】**显示防火墙规则的详细信息。

**【参数说明】**context:上下文;  
id:防火墙规则的 id;  
fields=None:字段。

**【返回值】**防火墙规则信息。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙规则信息。

**【提示】**获取成功/失败。

5) 更新防火墙规则方法:

**【方法原型】**update\_firewall\_rule(context, id, firewall\_rule)。

**【方法功能】**更新防火墙规则。

**【参数说明】**context:上下文;  
id:防火墙规则的 id;  
firewall\_rule:防火墙规则。

**【返回值】**防火墙规则字典。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙规则字典。

**【提示】**更新成功/失败。

6) 创建防火墙策略方法:

**【方法原型】**create\_firewall\_policy(context, firewall\_policy)。

**【方法功能】**创建一个防火墙策略。

**【参数说明】**context:上下文;  
firewall\_policy:防火墙策略。

**【返回值】**防火墙策略信息。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙策略信息。

**【提示】**创建成功/失败。

7) 删除防火墙策略方法:

**【方法原型】**delete\_firewall\_policy(context, id)。

**【方法功能】**删除一个指定的防火墙策略。

**【参数说明】**context:上下文;

id:防火墙策略的 id。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,删除防火墙策略。

**【提示】**删除成功/失败。

8) 添加防火墙规则方法:

**【方法原型】**insert\_rule(context, id, rule\_info)。

**【方法功能】**为防火墙策略添加一个防火墙规则。

**【参数说明】**context:上下文;

id:防火墙策略 id;

rule\_info 防火墙规则信息。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,添加防火墙规则。

**【提示】**添加成功/失败。

9) 获取防火墙策略方法:

**【方法原型】**get\_firewall\_policies(context, filters=None, fields=None)。

**【方法功能】**列出租户所有的全部防火墙策略。

**【参数说明】**context:上下文;

filters=None:过滤;

fields=None:字段。

**【返回值】**防火墙策略信息。

**【使用方法】**发起者调用该接口,传递参数,返回防火墙策略信息。

**【提示】**获取成功/失败。

10) 删除防火墙规则方法:

**【方法原型】**remove\_rule(context, id, rule\_info)。

**【方法功能】**为指定防火墙策略删除一条防火墙规则。

**【参数说明】**context:上下文;

id:防火墙策略 id;

rule\_info:防火墙规则信息。

**【返回值】**成功/失败。

**【使用方法】**发起者调用该接口,传递参数,删除防火墙规则。

**【提示】**删除成功/失败。

11) 获取防火墙策略方法:

**【方法原型】**get\_firewall\_policy(context, id, fields=None)。

**【方法功能】**显示所给出的防火墙策略的详细信息。

**【参数说明】**context:上下文;

id:防火墙策略 id;

fields=None:字段。

【返回值】防火墙策略信息。

【使用方法】发起者调用该接口,传递参数,获取防火墙策略信息。

【提示】获取成功/失败。

12) 更新防火墙策略方法:

【方法原型】update\_firewall\_policy(context, id, firewall\_policy)。

【方法功能】更新防火墙策略。

【参数说明】context:上下文;

id:防火墙策略 id;

firewall\_policy:防火墙策略。

【返回值】防火墙策略信息。

【使用方法】发起者调用该接口,传递参数,更新防火墙策略。

【提示】更新成功/失败。

### B.2.2 虚拟主机与应用服务器中间件接口

虚拟主机与应用服务器中间件接口的详细要求如下:

a) 软件包本地上传与下载方法:

1) 登录虚拟主机方法:

【方法原型】webssh2 (host, port, username, password)。

【方法功能】远程登录虚拟主机。

【参数说明】host:主机 IP 或域名;

port:ssh 端口;

username:登录名称;

password:登录密码。

【返回值】登录成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,登录虚拟主机。

【提示】Port:默认为 21 和 22,防火墙要开放此端口。

2) 上传文件方法:

【方法原型】upload (context, sourcefile, targetdiectory)。

【方法功能】上传文件到虚拟机指定目录。

【参数说明】context:登录信息上下文;

sourcefile:上传源文件;

targetdiectory:目的目录。

【返回值】上传成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,上传文件到虚拟主机的指定目录。

【提示】源文件可以是任何单文件。

3) 下载文件方法:

【方法原型】download (context, sourcefile)。

【方法功能】从远程虚拟主机上下载文件到本地。

【参数说明】context:登录信息上下文;

sourcefile:要下载的文件名称。

【返回值】下载成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,从远程虚拟主机上下载到本地目录。

【提示】无。

b) 软件包校验与防病毒方法:

1) 文件检验方法:

【方法原型】checksum (context, sourcefile, checksumcode)。

【方法功能】虚拟主机比较 checksumcode,完成软件包校验。

【参数说明】context:登录信息上下文;

sourcefile:要上传的文件名称;

checksumcode:上传前的 checksum。

【返回值】True/False。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,完成软件包校验

【提示】无。

2) 扫描方法:

【方法原型】security\_secribe (context, sourcefile)。

【方法功能】为上传的软件包实现防病毒扫描。

【参数说明】context:登录信息上下文;

sourcefile:要上传的文件名称。

【返回值】文件安全/不安全。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,扫描上传的文件。

【提示】无。

c) 软件包在线更新与下载方法:

【方法原型】install\_online (host, package)。

【方法功能】为虚拟主机提供在线包更新与安装功能。

【参数说明】host:在线包服务器地址;

package:要安装的或更新的软件包。

【返回值】更新、下载成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,在线下载、更新虚拟主机的软件包。

【提示】该方法仅适用于 linux 系列虚拟主机系统。

### B.2.3 虚拟主机与应用系统接口

虚拟主机与应用系统接口的详细要求如下:

a) 软件包提供本地上传与下载方法:

1) 登录方法:

【方法原型】webssh2 (host, port, username, password)。

【方法功能】远程登录虚拟主机。

【参数说明】host:主机 IP 或域名;

port:ssh 端口;

username:登录名称;

password:登录密码。

【返回值】登录成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,远程登录虚拟主机。

【提示】Port:默认为 21 和 22,防火墙要开放此端口。

2) 上传方法:

【方法原型】upload (context,sourcefile,targetdiectory)。

【方法功能】上传文件到虚拟机指定目录。

【参数说明】context:登录信息上下文;

sourcefile:上传源文件;

targetdiectory:目的目录。

【返回值】上传成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,上传文件到服务器。

【提示】源文件可以是任何单文件。

3) 下载方法:

【方法原型】download (context,sourcefile)。

【方法功能】从远程虚拟主机上下载文件到本地。

【参数说明】context:登录信息上下文;

sourcefile:要下载的文件名称。

【返回值】下载成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,从远程虚拟主机下载文件到本地。

【提示】无。

b) 软件包的提供校验与防病毒方法:

1) 检验方法:

【方法原型】checksum (context,sourcefile,checksumcode)。

【方法功能】虚拟主机比较 checksumcode,完成软件包校验。

【参数说明】context:登录信息上下文;

sourcefile:要上传的文件名称;

checksumcode:上传前的 checksum。

【返回值】True/False。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,校验软件包的完整性。

【提示】无。

2) 扫描方法:

【方法原型】security\_scribe (context,sourcefile)。

【方法功能】为上传的软件包实现防病毒扫描。

【参数说明】context:登录信息上下文;

sourcefile:要上传的文件名称。

【返回值】软件包安全/不安全。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,对上传的软件包进行病毒扫描。

【提示】无。

c) 在线迁移方法:

【方法原型】migrate\_online (context,uuid,dest\_host)。



【方法功能】为虚拟主机以及虚拟主机上的应用提供在线迁移功能。

【参数说明】context: 登录信息上下文;

uuid: 要迁移虚拟主机的 UUID;

dest\_host: 目标服务器。

【返回值】迁移成功/失败。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 迁移虚拟主机到其他物理主机上。

【提示】在线迁移应基于共享存储。

d) 负载均衡方法:

1) 创建资源池方法:

【方法原型】resource\_pool (name, description, provide, subnet, protols, method)。

【方法功能】增加一个负载均衡资源池。

【参数说明】name: 资源池名称;

description: 资源池描述;

provide: 提供者;

subnet: 子网;

protols: 协议;

method: 轮询方法。

【返回值】创建成功/失败。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 创建一个负载均衡资源池。

【提示】provide: http-proxy。

protols: http;

https。

method: round\_robin;

least\_connections;

source\_ip。

2) 添加主机方法:

【方法原型】addHost (name, member, weight, port)。

【方法功能】为资源池增加一个虚拟主机。

【参数说明】name: 资源池名称;

member: 主机数组;

weight: 权重;

port: 端口。

【返回值】添加成功/失败。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 为资源池增加一个虚拟主机。

【提示】无。

#### B.2.4 虚拟主机与存储设备接口

虚拟主机与存储设备接口的详细要求如下:

a) 存储设备的挂载与卸载方法:

1) 挂载方法:

【方法原型】attach\_volume(self, context, volume\_id, instance\_uuid, host\_name, mountpoint, mode)。

【方法功能】挂载云硬盘。

【参数说明】context:操作上下文;

volume\_id:云硬盘 id;

instance\_uuid:云主机 uuid;

host\_name:主机名;

mountpoint:挂载点;

mode:模式。

【返回值】挂载成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,将云硬盘挂载到虚拟主机。

【提示】无。

2) 卸载方法:

【方法原型】detach\_volume(context, volume\_id)。

【方法功能】卸载云硬盘。

【参数说明】context:操作上下文;

volume\_id:云硬盘 id。

【返回值】卸载成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,将云硬盘重虚拟主机卸载。

【提示】无。

b) 存储空间创建与销毁方法:

1) 创建方法:

【方法原型】create(self, req, body)。

【方法功能】创建一个云硬盘。

【参数说明】req:请求;

body:主体。

【返回值】包含云硬盘基本信息的字典对象{'volume': retval}。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,创建云硬盘。

【提示】无。

2) 删除方法:

【方法原型】delete(self, req, id)。

【方法功能】删除一个云硬盘。

【参数说明】req:请求;

id:云硬盘 id。

【返回值】删除成功/失败。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,删除云硬盘。

【提示】无。

3) 更新方法:

【方法原型】update(self, req, id, body)。

【方法功能】更新一个云硬盘。

**【参数说明】**req: 请求;  
id: 云硬盘 id;  
body: 主体。

**【返回值】**包含更新成功后的云硬盘详细信息的字典对象。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,更新云硬盘。

**【提示】**无。

4) 列出方法:

**【方法原型】**index(self, req)。

**【方法功能】**列出云硬盘总表。

**【参数说明】**req: 请求。

**【返回值】**包含云硬盘列表信息的字典对象。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,列出云硬盘列表信息。

**【提示】**无。

5) 详情方法:

**【方法原型】**detail(self, req)。

**【方法功能】**列出一个云硬盘列表详情。

**【参数说明】**req: 请求。

**【返回值】**包含云硬盘详细信息的字典对象。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,列出指定云硬盘的详细信息。

**【提示】**无。

c) 存储备份与恢复方法:

1) 创建方法:

**【方法原型】**create(self, req, body)。

**【方法功能】**创建一个新备份。

**【参数说明】**req: 请求;

body: 主体。

**【返回值】**备份成功后的备份对象的字典对象。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,创建存储备份。

**【提示】**无。

2) 删除方法:

**【方法原型】**delete(self, req, id)。

**【方法功能】**删除一个备份。

**【参数说明】**req: 请求;

id: 备份的 id。

**【返回值】**删除成功/失败。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,删除指定的备份。

**【提示】**无。

3) 恢复方法:

**【方法原型】**restore(self, req, id, body)。

**【方法功能】**恢复一个存在的备份云硬盘。

**【参数说明】**req: 请求;

id: 备份云硬盘的 id;

body: 主体。

【返回值】恢复成功后的备份对象的字典对象。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,恢复指定的备份。

【提示】无。

### B.3 数据库版本转换接口

数据库版本转换接口的详细要求如下:

a) 数据迁移转换方法:

【方法原型】data\_migrate\_sdb\_to\_tdb(sdb\_name, sdb\_version, sdb\_ip, sdb\_port, sdb\_dbname, sdb\_tablelist, sdb\_username, sdb\_pwd, tdb\_name, tdb\_version, tdb\_ip, tdb\_port, tdb\_dbname, tdb\_tablelist, tdb\_username, tdb\_pwd)。

【方法功能】实现相同数据库管理系统不同版本之间的数据迁移和转换,以及不同数据库管理系统之间的数据迁移和转换。

【参数说明】sdb\_name: 源数据库类型;

sdb\_version: 源数据库版本;

sdb\_ip: 源数据库 IP 地址;

sdb\_port: 源数据库服务端口;

sdb\_dbname: 源数据库名称;

sdb\_tablelist: 源数据库待迁移数据表列表;

sdb\_username: 源数据库访问用户名;

sdb\_pwd: 源数据库访问用户的密码;

tdb\_name: 目标数据库类型;

tdb\_version: 目标数据库版本;

tdb\_ip: 目标数据库 IP 地址;

tdb\_port: 目标数据库服务端口;

tdb\_dbname: 目标数据库名称;

tdb\_tablelist: 目标数据库数据迁移表列表;

tdb\_username: 目标数据库访问用户名;

tdb\_pwd: 目标数据库访问用户的密码。

【返回值】迁移转换结果。

【使用方法】接口运行在远程服务器上,发起者调用该接口,传递参数,进行数据迁移。

【提示】数据迁移和转换成功/失败。

b) SQL 适配和转换方法:

【方法原型】data\_sql\_adapter\_sdb\_to\_tdb(sdb\_name, sdb\_version, ssql, tdb\_name, tdb\_version)。

【方法功能】实现相同数据库管理系统不同版本之间的 SQL 适配和转换,以及不同数据库管理系统之间的 SQL 适配和转换。

【参数说明】sdb\_name: 源数据库类型;

sdb\_version: 源数据库版本;

ssql: 待适配和转换 SQL 语句;

tdb\_name: 目标数据库类型;

tdb\_version: 目标数据库版本。

【返回值】tsql: 适配和转换后的 SQL 语句。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 进行 SQL 转换和适配。

【提示】SQL 适配和转换成功/失败。

#### B.4 数据共享接口

数据共享接口的详细要求如下:

a) 目录注册方法:

【方法原型】directory\_register(token, metadata)。

【方法功能】实现目录元数据的采集和注册。

【参数说明】token: 用户认证 token;

metadata: 目录元数据和值。

【返回值】目录编号。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 进行目录采集和注册。

【提示】注册成功/失败。

b) 目录编目方法:

【方法原型】directory\_catalog(token, id, directory\_type\_id)。

【方法功能】将注册的元数据审核后并编目到某一目录分类中。

【参数说明】id: 目录编号;

directory\_type\_id: 目录类别编号;

token: 用户认证 token。

【返回值】成功/失败。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 进行目录审核和编目。

【提示】目录编目成功。

c) 目录发布方法:

【方法原型】directory\_publish(token, id)。

【方法功能】实现将某一个编目的目录进行发布, 使外部用户可以进行访问。

【参数说明】token: 用户认证 token。

id: 目录编号。

【返回值】成功/失败。

【使用方法】接口运行在远程服务器上, 发起者调用该接口, 传递参数, 进行目录发布。

【提示】发布成功。

d) 目录检索方法:

【方法原型】directory\_retrieval(token, id, directory\_name, directory\_type\_id, directory\_content)。

【方法功能】根据目录编号, 目录分类编号, 目录名称、目录内容检索目录信息, 目录名称、目录内容检索支持模糊查询, 并返回检索到的目录列表。除用户认证 token 外, 其他参数任选其一。

【参数说明】token: 用户认证 token;

id: 目录编号;

directory\_name: 目录名称;

directory\_type\_id: 目录分类编号;

directory\_content: 目录内容。

**【返回值】**目录详细信息/目录编号列表。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,进行目录检索。

**【提示】**目录检索成功。

e) 数据交换方法:

**【方法原型】**data\_exchange(token,srouter\_id, data,trouter\_id\_list)。

**【方法功能】**实现数据的交换,支持结构化和非结构化数据的交换,支持一对一、一对多。

**【参数说明】**token:用户认证 token;

srouter\_id:数据交换发送方路由编号;

data:待交换的数据;

trouter\_id\_list:数据交换接收方路由编号,可以是一个或者多个。

**【返回值】**成功/失败。

**【使用方法】**接口运行在远程服务器上,发起者调用该接口,传递参数,进行数据交换。

**【提示】**数据交换成功/失败。

## B.5 应用系统与应用服务器中间件接口

应用系统与应用服务器中间件接口的详细要求如下:

a) 应用打包方法:

**【方法原型】**export\_app(source\_target,dest\_target,level)。

**【方法功能】**提供应用系统之间同步数据发送。

**【参数说明】**source\_target:需要进行打包的系统描述数据体;

dest\_target:需要转换的目标部署系统描述数据体;

level:转换级别。

**【返回值】**转换结果。

**【使用方法】**需要进行部署环境转换时,该接口被调用,输入相应参数,完成业务系统在不同中间件中的部署转换。

**【提示】**无。

b) 应用部署方法:

**【方法原型】**deploy\_app(target,app,deploy\_content)。

**【方法功能】**提供应用在不同应用服务器中间件中部署的方法。

**【参数说明】**target:系统的目标部署环境描述数据体;

app:需要转换的目标部署系统描述数据体;

deploy\_content:部署参数数据体。

**【返回值】**部署结果。

**【使用方法】**需要进行部署时,该接口被调用,输入相应参数,完成业务系统的部署。

**【提示】**无。

c) 应用迁移方法:

**【方法原型】**migration\_app(source\_target,dest\_target,deploy\_content)。

**【方法功能】**提供应用系统之间同步数据发送。

**【参数说明】**source\_target:需要进行迁移的系统描述数据体;

dest\_target:需要转换的目标部署系统描述数据体;

deploy\_content:迁移参数数据体。

**【返回值】**迁移结果。

**【使用方法】**需要进行应用迁移转换时,该接口被调用,输入相应参数,完成业务系统在不同中间件中的迁移。

**【提示】**无。

d) 应用监控方法:

**【方法原型】**monitor\_app(target\_app, app\_adapter, monitor\_content)。

**【方法功能】**提供应用系统在不同应用服务器中间件环境中管理,监控适配的方法。

**【参数说明】**target\_app:需要进行监控的系统描述数据体;

app\_adapter:业务系统监控管理适配器;

monitor\_content:监控、管理参数数据体。

**【返回值】**适配结果。

**【使用方法】**需要进行应用管理、监控适配时,该接口被调用,输入相应参数,完成业务系统在不同中间件中的业务系统的监控、管理适配操作。

**【提示】**无。

e) 服务调用方法:

**【方法原型】**service\_brtidge (souce\_service, target\_service, transform\_content)。

**【方法功能】**提供应用系统在不同平台,不同开发语言之间进行服务调用方法。

**【参数说明】**souce\_service:源业务系统服务描述数据体;

target\_service:目标业务系统服务描述数据体;

transform\_content:接口转换参数描述数据体。

**【返回值】**服务接口转换结果。

**【使用方法】**需要进行服务接口转换时,该接口被调用,输入相应参数,完成跨平台,异构语言之间服务调用转换。

**【提示】**无。

## B.6 应用服务器互操作接口

无示例。

## B.7 应用系统互操作调用接口

应用系统互操作调用接口的详细要求如下:

a) 同步消息发送方法:

**【方法原型】**send\_message\_sync(remote\_ip, remote\_port, content)。

**【方法功能】**提供应用系统之间同步数据发送。

**【参数说明】**remote\_ip:远程应用 IP;

remote\_port:远程应用服务端口;

content:传递的调用信息。

**【返回值】**数据发送成功或失败。

**【使用方法】**调用者实现该接口,向目标应用系统发送调用信息。

**【提示】**无。

b) 同步消息接收方法:

**【方法原型】**receive\_message\_sync(local\_port, content)。

**【方法功能】**提供应用系统之间同步接收数据;

**【参数说明】**local\_port:本地消息监听端口;

content:传递的调用信息。

**【返回值】**数据发送成功或失败。

**【使用方法】**消息接收者实现该接口,收取发送的信息。

**【提示】**无。

c) 异步消息发送方法:

**【方法原型】**send\_message\_async(remote\_ip, remote\_port, content)。

**【方法功能】**提供应用系统之间异步数据发送。

**【参数说明】**remote\_ip:远程应用 IP;

remote\_port:远程应用服务端口;

content:传递的调用信息。

**【返回值】**数据发送成功或失败。

**【使用方法】**调用者实现该接口,向目标应用系统发送调用信息。

**【提示】**无。

d) 异步消息接收方法:

**【方法原型】**receive\_message\_async(local\_port, content)。

**【方法功能】**提供应用系统之间异步接收数据。

**【参数说明】**local\_port:本地消息监听端口;

content:传递的调用信息。

**【返回值】**数据发送成功或失败。

**【使用方法】**消息接收者实现该接口,收取发送的信息。

**【提示】**无。

e) 用户名,口令认证方法:

**【方法原型】**authenticate\_user(user\_id, passwd)。

**【方法功能】**用户使用用户名,口令进行用户身份认证。

**【参数说明】**user\_id:用户唯一识别身份标识;

passwd:用户身份口令。

**【返回值】**认证成功或认证失败。

**【使用方法】**该接口运行在远程服务器上,认证发起者调用该接口,传递参数,获取认证结果。

**【提示】**无。

f) 证书认证方法:

**【方法原型】**authenticate\_cert(cert\_id, cert\_pin)。

**【方法功能】**用户数字证书进行用户身份认证。

**【参数说明】**cert\_id:数字证书识别码;

cert\_pin:数字证书 PIN 码。

**【返回值】**认证成功或认证失败。

**【使用方法】**该接口运行在远程服务器上,认证发起者调用该接口,传递参数,获取认证结果。

**【提示】**无。

## B.8 公共平台资源服务管理接口

公共平台资源服务管理接口的详细要求如下:

a) 用户名,口令认证方法:



**【方法原型】**authenticate\_user(user\_id,passwd)。

**【方法功能】**用户使用用户名,口令进行用户身份认证。

**【参数说明】**user\_id:用户唯一识别身份编码;

passwd:用户身份口令。

**【返回值】**认证成功或认证失败。

**【使用方法】**该接口运行在远程服务器上,认证发起者调用该接口,传递参数,获取认证结果。

**【提示】**无。

b) 证书认证方法:

**【方法原型】**authenticate\_cert(cert\_id,cert\_pin)。

**【方法功能】**用户数字证书进行用户身份认证。

**【参数说明】**cert\_id:数字证书识别码;

cert\_pin:数字证书 PIN 码。

**【返回值】**认证成功或认证失败。

**【使用方法】**该接口运行在远程服务器上,认证发起者调用该接口,传递参数,获取认证结果。

**【提示】**无。

c) 授权方法:

**【方法原型】**resources\_authorize (user\_id,zone\_id)。

**【方法功能】**获取用户在一个域内的授权可访问资源的信息。

**【参数说明】**user\_id:用户唯一识别身份编码;

zone\_id:资源域编码。

**【返回值】**用户在该域内的授权可访问资源信息。

**【使用方法】**该接口运行在远程服务器上,认证发起者调用该接口,传递参数,获取用户在该域内的授权可访问资源信息。

**【提示】**无。

d) 获取计算资源平台地址:

**【方法原型】**getComputerPlatform()。

**【方法功能】**该方法获取计算资源平台地址。

**【参数说明】**无。

**【返回值】**计算资源平台链接地址。

**【使用方法】**获取数据中心计算资源平台的地址。

**【提示】**无。

e) 获取数据库资源平台地址方法:

**【方法原型】**getDBPlatform()。

**【方法功能】**该方法获取数据库资源平台地址。

**【参数说明】**无。

**【返回值】**数据库资源平台链接地址。

**【使用方法】**获取数据中心数据库资源平台的地址。

**【提示】**无。

f) 获取数据共享资源平台地址方法:

**【方法原型】**getStoragePlatform()。

**【方法功能】**该方法获取数据共享资源平台地址。

**【参数说明】**无。

**【返回值】**数据共享资源平台链接地址。

**【使用方法】**该方法获取数据中心数据共享资源平台的地址。

**【提示】**无。

g) 获取中间件资源平台地址方法：

**【方法原型】**get MiddlewarePlatform()。

**【方法功能】**该方法获取中间件资源平台地址。

**【参数说明】**无。

**【返回值】**中间件资源平台链接地址。

**【使用方法】**获取数据中心计算中间件平台的地址。

**【提示】**无。

h) 应用系统推送监控数据方法：

**【方法原型】**sendAppMonitorMessage(IPAddress,monitorMessage)。

**【方法功能】**该方法负责将应用系统监控数据推送到平台。

**【参数说明】**IPAddress:平台服务 IP 地址；

monitorMessage:应用系统监控数据报文。

**【返回值】**无。

**【使用方法】**应用系统推送监控数据时调用该方法。

**【提示】**无。

i) 平台接收应用系统声明报文发送方法：

**【方法原型】**receiveDeclare()。

**【方法功能】**该方法用于接收各应用系统发送的监控指标的声明报文。

**【参数说明】**无。

**【返回值】**无。

**【使用方法】**平台接收应用系统声明报文时调用。

**【提示】**无。

j) 平台接收应用系统监控数据方法：

**【方法原型】**receiveMonitor()。

**【方法功能】**该方法用于接收各应用系统发送的监控数据。

**【参数说明】**无。

**【返回值】**无。

**【使用方法】**平台接收应用系统监控数据时调用。

**【提示】**无。

---



中华人民共和国  
国家标准  
基于云计算的电子政务公共平台技术  
规范 第3部分：系统和数据接口  
GB/T 33780.3—2017

\*

中国标准出版社出版发行  
北京市朝阳区和平里西街甲2号(100029)  
北京市西城区三里河北街16号(100045)

网址 [www.spc.net.cn](http://www.spc.net.cn)

总编室：(010)68533533 发行中心：(010)51780238

读者服务部：(010)68523946

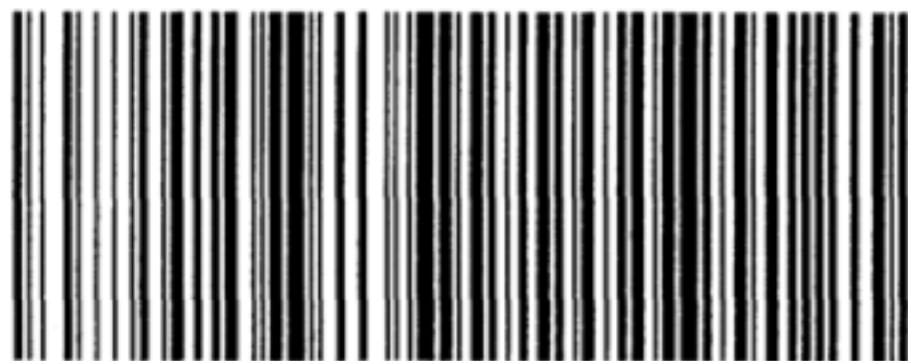
中国标准出版社秦皇岛印刷厂印刷  
各地新华书店经销

\*

开本 880×1230 1/16 印张 2.75 字数 74 千字  
2017年8月第一版 2017年8月第一次印刷

\*

书号：155066·1-56394 定价 39.00 元



GB/T 33780.3—2017