

T/BAX

团 体 标 准

T/BAX XXXX.5—2023

视频图像感知智能应用适配技术要求  
第5部分：算法算力服务管理平台  
技术要求

Technical requirements for intelligent application adaptation of video and image sensing—Part 5: Technical requirements for algorithmic computing power service management

(征求意见稿)

XXXX-XX-XX 发布

XXXX-XX-XX 实施

# 目 次

前 言 .....	III
引 言 .....	IV
1 范围 .....	1
2 规范性引用文件 .....	1
3 术语和定义 .....	1
3.1 术语和定义 .....	1
3.2 缩略语 .....	1
4 功能框架 .....	1
5 算法接入 .....	2
5.1 平台算法包管理 .....	2
5.2 平台算法注册管理 .....	2
6 算法管理 .....	2
6.1 边端设备算法管理 .....	2
6.2 中心算力算法管理 .....	2
6.3 任务控制 .....	2
6.4 算法调度 .....	3
6.5 算法评估 .....	3
6.6 算法调度流程 .....	3
6.7 算法鉴权 .....	4
7 算力接入 .....	5
7.1 设备算力接入 .....	5
7.2 中心算力接入 .....	5
8 算力管理 .....	5
8.1 设备算力管理 .....	6
8.2 中心算力监控 .....	6
8.3 中心算力调度 .....	6
8.4 中心计算平台集群管理 .....	6
8.5 工作负载管理 .....	6
8.6 算力资产管理 .....	6
9 视图连接要求 .....	6
9.1 视图接入功能要求 .....	6
9.2 视频设备管理功能要求 .....	6
10 数据共享要求 .....	6

10.1	数据服务管理功能要求	6
10.2	数据资源管理功能要求	6
10.3	数据交换管理功能要求	6
附录 A	(规范性) 算法管理接口	7
附录 B	(规范性) 边端算法包部署	26
附录 C	(规范性) 算法算力服务管理平台任务调度	34
附录 D	(规范性) 鉴权要求	72
附录 E	(规范性) 设备算力接入	83
附录 F	(规范性) 中心算力查询接口	88
附录 G	(规范性) 边端设备鉴权接口	96
附录 H	(规范性) 中心鉴权服务接口	99
附录 I	(规范性) 加速卡类型	101
参 考 文 献		102

## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是T/BAX XXXX-2023《视频图像感知智能应用适配技术要求》的第5部分。

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专利的责任。

本文件由北京安全防范行业协会提出并归口。

本文件起草单位：北京市公安局、北京市经济和信息化局、视频图像信息智能分析与共享应用技术国家工程实验室、北京百度网讯科技有限公司、北京旷视科技有限公司、富盛科技股份有限公司、北京太初元芯集成电路有限公司、北京瑞莱智慧科技有限公司。

本文件主要起草人：陶山、步飞、崔云红、赵惠芳、王建勇、苑辰、王卿鹏、王岭钢、张伟、钟永强、王洪磊、张浩天。

## 引 言

视频图像感知智能应用是国家数字化进程和智慧城市建设中的重要内容，在公共安全管理、社会治理、社区管理等应用场景中，发挥关键作用。

T/BAX XXXX-202X《视频图像感知智能应用适配技术要求》旨在通过建立统一的适配技术规范，指导视频图像解析算法与设备的适配，解决算法与设备的绑定问题，实现算法随业务需求持续快速更新迭代，并通过云边端算力动态调度，逐步实现算法快速部署。

T/BAX XXXX-202X 拟由五部分构成。

- 第1部分：总体技术要求。目的在于确定视频图像感知智能化应用适配的总体结构，以及适配需要遵循的总体要求和具体要求等技术要求。
- 第2部分：算法适配技术要求。目的在于规范视频图像感知智能化应用中算法适配的基本要求、功能要求、算法授权要求、接口要求等技术要求。
- 第3部分：感知前端和边缘解析设备适配技术要求。目的在于规范视频图像感知智能化应用中感知前端和边缘解析设备的基本要求、功能要求、接口要求等技术要求。
- 第4部分：中心算力适配技术要求。目的在于规范视频图像感知智能化应用中中心算力的总体框架、功能要求、接口要求等技术要求。
- 第5部分：算法算力服务管理平台技术要求。目的在于规范视频图像感知智能化应用中算法接入、算法管理、算力接入、算力管理、视频图像接入、数据共享等技术要求。

# 视频图像感知智能应用适配技术要求

## 第5部分：算法算力服务管理平台技术要求

### 1 范围

本文件规定了视频图像感知智能应用适配算法算力服务管理平台的功能框架、算法接入、算法管理、算力接入、算力管理，视频图像连接要求和数据共享要求。

本文件适用于视频图像感知智能应用适配算法算力服务管理平台的设计、开发、检测和验收。

### 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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。

GA/T 1399.1—2017 公安视频图像分析系统 第1部分：通用技术要求

GA/T 1400.1—2017 公安视频图像信息应用系统 第1部分：通用技术要求

T/BAX XXXX.1-202X 视频图像感知智能应用适配技术要求 第1部分：总体技术要求

T/BAX XXXX.2-202X 视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求

T/BAX XXXX.4-202X 视频图像感知智能应用适配技术要求 第4部分：中心算力适配技术要求

### 3 术语和定义

#### 3.1 术语和定义

GA/T 1399.1—2017、GA/T 1400.1—2017、T/BAX XXXX.1-202X界定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。

ACMP：算法算力服务管理平台（Algorithm Computing Management Platform）

### 4 功能框架

视频图像感知智能应用算法算力服务管理平台用于对中心算力、感知前端和边端解析设备（以下简称“边端设备”）等算法和算力的统一管理，其功能框架如图1所示，主要包括算法接入、算法管理、算力接入、算力管理，以及视图连接和数据共享等功能模块。

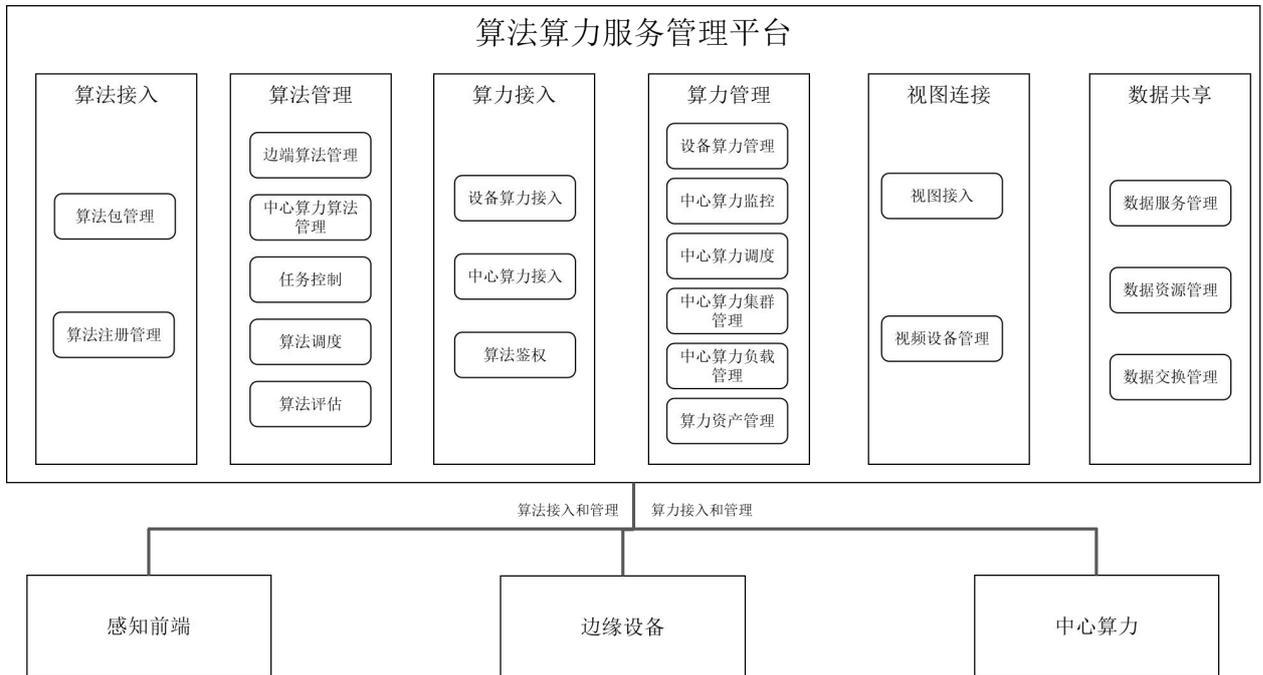


图1 视频图像感知智能应用算法算力服务管理平台功能框架

## 5 算法接入

### 5.1 平台算法包管理

算法包或算法引擎导入前，应遵循T/BAX XXXX. 2-202X（《视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求》）中的规定进行封装。

算法算力服务管理平台应提供算法包管理接口，支持算法包的导入、删除和信息查询，接口要求见附录A. 2-A. 5。

### 5.2 平台算法注册管理

算法算力服务管理平台应支持算法的注册管理，包括对算法的创建、更新、查询和删除等，接口要求见附录A. 6-A. 10。

## 6 算法管理

### 6.1 边端设备算法管理

算法算力服务管理平台应支持向边端设备下发、删除、安装、卸载，以及查询算法包，管理边端所运行的算法包，接口要求见附录B。

### 6.2 中心算力算法管理

算法算力服务管理平台应支持对中心算力的算法引擎或算法包的管理，包括算法引擎或算法包的创建、删除、更新和查询。此外平台应支持对中心算力上算法运行的控制，包括算法服务的删除、更新和查询。中心算力镜像和算法服务的下发和启停由任务调度内部驱动调用。

### 6.3 任务控制

算法算力服务管理平台应支持以下任务控制能力：

- a) 支持对解析任务进行批量启停控制，并提供多种任务运行方式，如周期、巡检等任务模式，以及多种形式的告警方式，相关接口要求见附录 C。
- b) 支持对中心算力运行算法进行控制，具备配置任务实例并进行启停控制的能力，相关接口应遵循 T/BAX XXXX. 2-202X（《视频图像感知智能应用适配技术要求 第 2 部分：算法适配技术要求》）；
- c) 支持对各边端设备配置任务实例并进行启停控制的能力，http 模式接口见 T/BAX XXXX. 2-202X，SIP 模式接口见 T/BAX XXXX. 3-202X（《视频图像感知智能应用适配技术要求 第 3 部分：感知前端和边缘解析设备适配技术要求》）中的要求；
- d) 支持订阅中心计算及边端设备解析任务所产生数据，数据回调接口见 T/BAX XXXX. 2-202X（《视频图像感知智能应用适配技术要求 第 2 部分：算法适配技术要求》）。

#### 6.4 算法调度

平台支持部署调度、扩缩容调度、任务分配调度等调度策略，并依据策略配置算力资源，具体包括：

- a) 应支持对所有用户的算法需求，进行统一部署调度；
- b) 应支持根据算法引擎的分析类型进行部署调度；
- c) 应支持根据视图数据集群的网络情况进行手动扩缩容；
- d) 应支持基于算法引擎部署情况，对计算任务均匀分配调度；
- e) 宜支持选择最近算力进行自动扩缩容调度；
- f) 宜支持基于集群算力分布、资源使用等状态的自动扩缩容调度；
- g) 宜支持算法编排的能力。

#### 6.5 算法评估

平台提供算法测试准入、算法比对和应用效果等的评估，具体包括：

- a) 应支持算法准入测试，验证算法所声明的效果和性能的真实性；
- b) 应支持多个同类算法真实运行数据的无标注自动比对；
- c) 宜支持对算法实际运行所需资源、运行状态等的评估。

#### 6.6 算法调度流程

平台算法调度流程如图2所示。

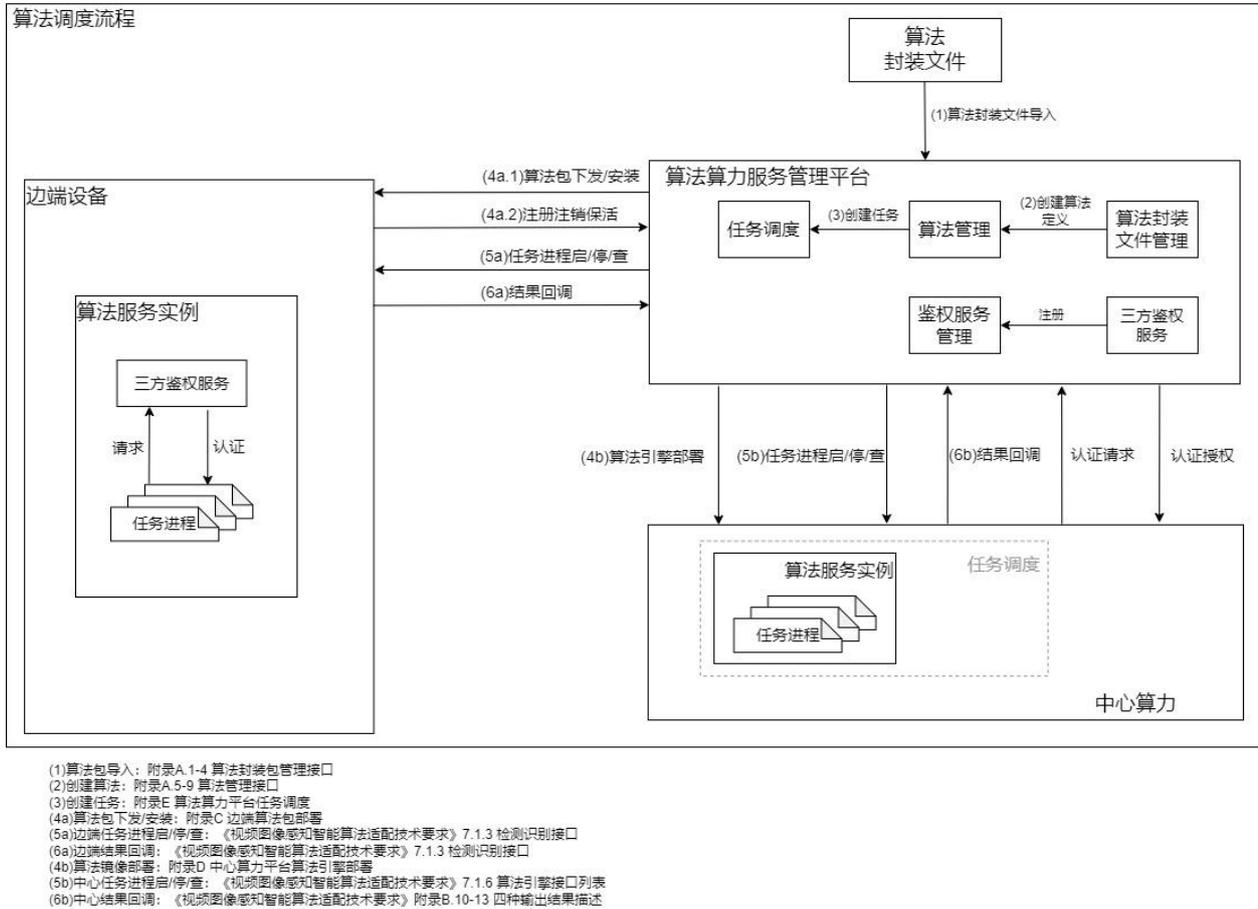


图 2 算法调度流程图

其中：

流程 (1)：算法包导入到算法算力服务管理平台，接口要求遵循附录A. 2至附录A. 5中的规定；

流程 (2)：算法算力服务管理平台基于算法包信息创建算法定义，接口要求遵循附录A. 6至附录A. 10中的规定；

流程 (3)：算法算力服务管理平台基于算法创建任务以便执行任务，接口要求遵循附录E中的规定；

流程 (4a)：算法包下发/安装至边缘设备，接口要求遵循附录B中的规定；

流程 (5a)：平台对边缘任务进程启/停/查，接口要求遵循T/BAX XXXX. 2—202X（《视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求》）中7.1.3 的规定；

流程 (6a)：边缘结果回调至平台，接口要求遵循T/BAX XXXX. 2—202X（《视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求》）中7.1.3 的规定；

流程 (4b)：算法引擎部署至中心计算平台，接口要求见遵循T/BAX XXXX. 4—202X（《视频图像感知智能应用适配技术要求 第4部分：中心算力适配技术要求》）中附录B的规定；

流程 (5b)：算法算力服务管理平台对中心任务进程启/停/查，接口要求见遵循T/BAX XXXX. 2—202X（《视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求》）中7.1.6的规定；

流程 (6b)：中心计算的结果回调至算法算力服务管理平台，接口要求遵循T/BAX XXXX. 2—202X（《视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求》）中附录B. 10-B. 13的规定。

### 6.7 算法鉴权

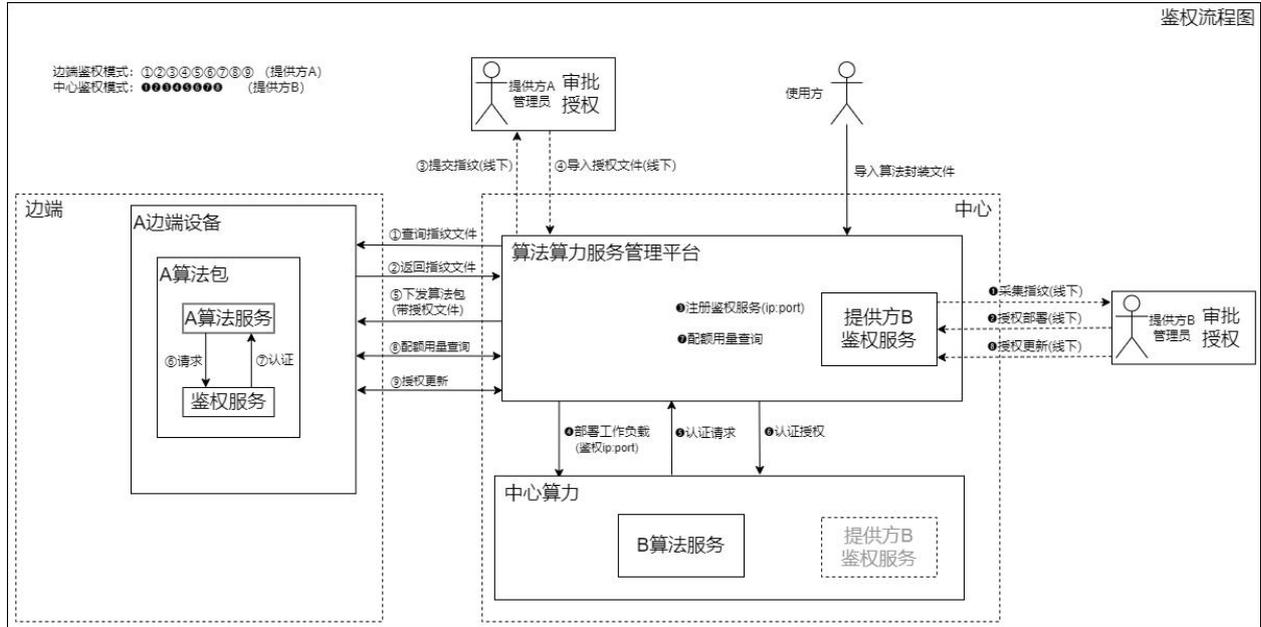


图3 算法鉴权流程图

算法鉴权流程如图3所示，鉴权分为两种模式，相关要求如下：

- 中心鉴权模式，算法提供方部署鉴权服务器，线下完成鉴权指纹采集和授权工作，将鉴权服务地址注册到算法算力服务管理平台，平台部署算法服务时作为参数传入鉴权服务地址；
- 边端鉴权模式，边端设备提供机器指纹查询接口供算法算力服务管理平台查询，线下获取机器的授权文件后导入平台，平台将鉴权文件随算法包下发到边端设备；
- 算法包和鉴权/授权服务通过 provider 提供方字段关联部署在中心算力的算法使用“中心鉴权模式”，部署在边端设备的算法使用“边端鉴权模式”；
- 平台鉴权相关接口应遵循附录D的规定，边端和算力中心鉴权接口应遵循附录G和附录H的规定。

## 7 算力接入

### 7.1 设备算力接入

设备算力接入时，平台应支持计算设备的统一注册、注销、保活、状态和资源查询等，相关接口见附录F。

### 7.2 中心算力接入

中心算力接入时，平台应满足以下要求：

- 平台应支持中心计算平台接入，以及集群和节点的创建、删除、查询等，对中心计算平台集群查询接口要求见附录F；
- 宜支持对中心解析设备所提供通用CPU机器、通用GPU机器及机器资源的统一注册；
- 宜支持对中心解析平台负载、集群注册与查询、算力监控的连接、适配与服务接口管理与调用；
- 宜支持vGPU的接入。

## 8 算力管理

## 8.1 设备算力管理

平台具备对设备算力进行管理的功能，应满足以下要求：

- a) 支持边端计算设备的物理状态及算力资源使用等信息查询，包括节点、设备和卡、算法的实时查询；
- b) 支持计算设备注销；
- c) 宜支持计算设备的负载注册、注销和保活，应支持计算设备服务进程信息注册到平台，包括服务地址等信息；

## 8.2 中心算力监控

平台具备对中心算力的监控功能，应满足以下要求：

- a) 支持对多中心算力资源使用的实时监控；
- b) 支持节点及 AI 加速卡级别的算力监控，如 AI 卡、CPU、显存使用率实时信息及历史趋势；
- c) 支持 vGPU 及所承载 AI 服务实例的 GPU、显存、CPU 用量实时监控。

## 8.3 中心算力调度

平台具备算力调度功能，应满足以下要求：

- a) 支持向中心算力平台申请算法所需算力资源；
- b) 支持配置多平台算力调度策略，可根据各算力平台资源情况自动或手动完成负载部署；
- c) 支持查询多算力平台已部署的负载服务；
- d) 支持算法在各算力平台使用算力资源监控，包括平台、实例数、AI 算力、显存、CPU 信息。

## 8.4 中心计算平台集群管理

平台具备集群管理能力，应满足以下要求：

- a) 支持多中心计算平台集群统一查询，查询信息包括集群操作状态、运行状态、集群资源；
- b) 支持扩容和缩容算力集群节点，支持对已有负载节点的扩缩容校验；
- c) 支持查询集群数、节点数、服务数及 各类异构 AI 加速卡、CPU、内存总量、分配量及使用；
- d) 支持边缘计算设备节点注册及边缘算力集群管理。

## 8.5 工作负载管理

平台具备对中心算力的负载管理能力，应满足以下要求：

- a) 支持多中心算力平台的统一负载管理和算法服务可用性校验；
- b) 支持通过应用算力需求额度，查询对该资源需求资源分配最合理的中心算力平台集群；
- c) 支持查询工作负载列表和实例列表，包括 AI 加速卡类型及数量、CPU、内存资源占用等；
- d) 支持将适配不同 AI 硬件的 AI 服务由算法算力服务管理平台调度部署到对应类型的 AI 算力节点；
- e) 支持应用服务通过集群应用服务网关提供外部访问能力；
- f) 支持服务列表管理，支持查看服务所关联的 pod 数量、服务集群内部地址以及集群外部访问地址。

## 8.6 算力资产管理

平台宜具备算力资产管理能力，包括：

- a) 管理算力申请、分配、回收过程，保障各账户算力资产及权限隔离；
- b) 算力总量、分配、使用运行等管控数据统计。

## 9 视图连接要求

### 9.1 视图接入功能要求

平台应支持视图接入，应满足以下要求：

- a) 支持符合 GB/T 28181-2022 要求的设备管理平台的注册及接入，支持用户通过界面查询已注册的下级平台列表；
- b) 支持符合 GB/T 28181-2022 要求的设备管理平台的注册、接入及查询，包括国标 ID、名称、父节点 ID、在线状态；
- c) 支持符合 GB/T 28181-2022 要求的设备批量导入操作，支持用户选择批量设备完成设备导入；
- d) 支持 GA/T 1400 协议的视图库接入。

## 9.2 视频设备管理功能要求

平台支持视频设备管理，应满足以下要求：

- a) 支持视频设备节点的注册、修改、删除、查看；
- b) 支持视频设备的注册、包括国标 ID、视频流地址、经纬度；
- c) 支持按照视频设备名称、设备标签、设备类型进行设备筛选；
- d) 支持查询正在运行的 AI 分析任务实例数及任务列表；
- e) 支持对视频设备关联的所有算法进行详情查询；
- f) 支持 GB/T 28181-2022 视频流的播放订阅等。

## 10 数据共享要求

### 10.1 数据服务管理功能要求

平台提供数据服务管理功能，应包括：

- a) 第三方数据 API 接入配置，支持手动配置 API 输入/输出参数/权属信息等；
- b) 数据源创建、编辑、查询、删除数据服务 API；
- c) 平台内创建数据服务的拨测；
- d) 宜支持接口文档自动生成。

### 10.2 数据资源管理功能要求

平台提供数据资源管理功能，应支持数据库连接管理和测试，以及第三方数据平台库表的接入和信息配置。

### 10.3 数据交换管理功能要求

平台提供数据交换管理功能，应包括：

- a) 数据交换任务的查看、启停控制、任务删除；
- b) 交换数据的追加、覆盖与更新；
- c) 交换任务执行日志的查询。

**附录 A**  
**(规范性)**  
**算法管理接口**

**A.1 基本要求**

本章节共规定了9个算法接口，主要包括算法封装文件和算法的导入、删除、创建、更新等，见表A.1。其中，算法算力服务管理平台（以下简称“算法算力平台”）与用户之间的算法接口调度流程见图A.1，各个接口属性信息应符合表A.2-A.10的规定，具体应符合章节A.2-A.10的规定。

**表 A.1 算法包管理接口**

序号	URI	方法	功能说明
1	/ACP/algoFilePackages	POST	导入平台算法封装文件
2	/ACP/algoFilePackages/{id}	DELETE	删除平台算法封装文件
3	/ACP/algoFilePackages	GET	查询平台算法封装文件列表
4	/ACP/algoFilePackages/{id}	GET	查询平台算法封装文件详情
5	/ACP/algoDef	POST	创建算法能力
6	/ACP/algoDef/{code}	PUT	更新算法能力
7	/ACP/algoDef	GET	查询算法能力列表
8	/ACP/algoDef/{code}	GET	查询算法能力详情
9	/ACP/algoDef/{code}	DELETE	删除算法能力

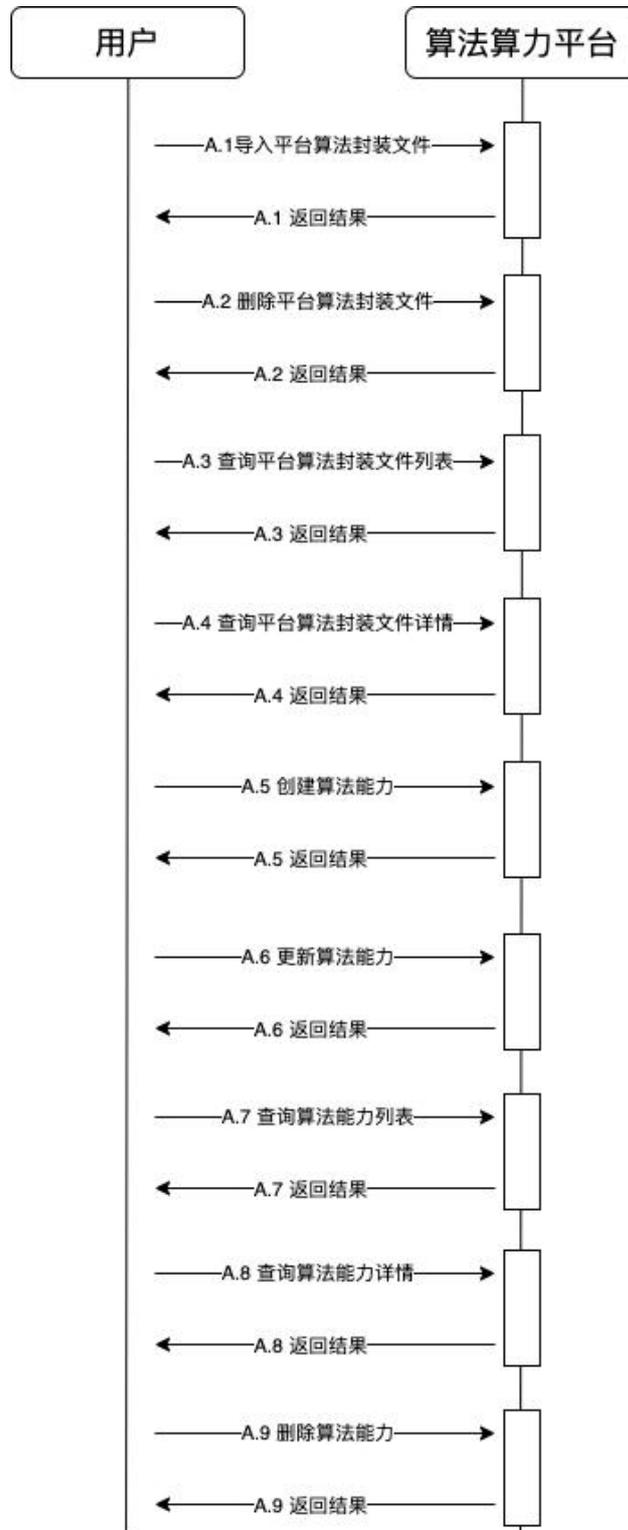


图 A. 1 算法管理时序图

表 A. 2 导入平台算法封装文件包

URI	/ACP/algoFilePackages
-----	-----------------------

功能	导入平台算法封装		
方法	查询字符串	请求消息体	响应消息体
POST	无	FILE	EmptyResponse
注释	<p>算法封装文件导入后平台生成算法封装文件ID</p> <p>对于引擎类算法封装文件,导入后算法算力服务管理平台需解析描述文件创建算法引擎,生成engineCode算法引擎编码,然后解析算法能力描述生成算法定义。</p> <p>对于算法包类型的算法封装文件包,导入后算法算力服务管理平台需解析描述文件生成算法包信息。算法封装文件描述文件中应包含每个算法能力/算法包的编码,算法能力/算法包编码由用户自定义,需要唯一,建议格式:provider-算法英文-version,由大小写字母、下划线、横线、英文.组成,长度小于128字符,例如 BAIDU-HUMAN_ATTR_ALERT-1.0.0。</p> <p>导入的算法能力/算法包编码如果已存在会执行覆盖更新逻辑,如果有算法服务在运行中,导入会报错。</p>		

表 A.3 删除平台算法封装文件包

URI	/ACP/algoFilePackages/{id}		
功能	删除平台算法封装文件		
方法	查询字符串	请求消息体	响应消息体
DELETE	无	无	EmptyResponse
注释	根据算法封装文件ID从算法算力服务管理平台删除算法封装文件包		

表 A.4 查询平台算法封装文件包列表

URI	/ACP/algoFilePackages		
功能	查询平台算法封装文件列表		
方法	查询字符串	请求消息体	响应消息体
GET	无	AlgoFilePackageListRequest	AlgoFilePackageListResponse
注释	查询平台算法封装文件列表		

表 A.5 查询平台算法封装文件包详情

URI	/ACP/algoFilePackages/{id}		
功能	查询平台算法封装文件详情		
方法	查询字符串	请求消息体	响应消息体
GET	无	无	AlgoFilePackageDetailResponse
注释	根据算法封装文件ID从算法算力服务管理平台查询详情		

表 A.6 创建算法能力

URI	/ACP/algoDef		
功能	创建算法能力		
方法	查询字符串	请求消息体	响应消息体
POST	无	TaskDefCreateRequest	EmptyResponse
注释	<p>算法封装文件描述文件中应包含每个算法能力的编码,算法能力编码由用户自定义,需要唯一,建议格式:provider-算法英文-version,由大小写字母、下划线、横线、英文.组成,长度小于128字符,例如 BAIDU-HUMAN_ATTR_ALERT-1.0.0。</p>		

表 A.7 更新算法能力

URI	/ACP/algoDef/{code}		
功能	更新算法能力		
方法	查询字符串	请求消息体	响应消息体
PUT	无	TaskDefUpdateRequest	EmptyResponse
注释	已下发或已部署的算法能力无法更新		

表 A.8 查询算法能力列表

URI	/ACP/algoDef		
功能	查询算法能力列表		
方法	查询字符串	请求消息体	响应消息体
GET	无	TaskDefPageRequest	TaskDefPageResponse
注释	查询算法能力列表		

表 A.9 查询算法能力详情

URI	/ACP/algoDef/{code}		
功能	查询算法能力详情		
方法	查询字符串	请求消息体	响应消息体
GET	无	code	TaskDefDetailResponse
注释	查询算法能力详情		

表 A.10 删除算法能力

URI	/ACP/algoDef/{code}		
功能	删除算法能力		
方法	查询字符串	请求消息体	响应消息体
DELETE	无	code	EmptyResponse
注释	删除算法能力		

## A.2 导入平台算法封装文件

- a) URI: POST/ACP/algoFilePackages
- b) 请求参数

表 A.11 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件	file	MultipartFile		R	算法 tar 包文件, 使用 multipart/form-data 格式上传

- c) 响应参数:

表 A.12 EmptyResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	ID		R	数据
4	请求 id	requestId	String		0	请求 ID

表 A.13 ID 信息

序号	名称	标识符	类型	长度	必选/可选	描述
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	描述
1	编号	id	Integer		R	算法封装文件 ID

d) 请求示例

Content-Type: multipart/form-data;

file: algoPackage.tar

e) 响应示例

```
{
  "code": "0",
  "message": "success",
  "data": {
    "id": 1
  }
}
```

### A.3 删除平台算法封装文件

a) URI: DELETE /ACP/algoFilePackages/{id}

b) 请求参数:

表 A.14 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件 ID	id	Long		R	算法封装文件 ID, 导入算法封装文件后由平台生成

c) 响应参数:

EmptyResponse

d) 请求示例:

DELETE /ACP/algoFilePackages/1

e) 响应示例:

```
{
  "code": "0",
  "message": "success"
}
```

### A.4 查询平台算法封装文件列表

a) URI: GET /ACP/algoFilePackages

b) 请求参数:

表 A.15 AlgoFilePackageListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件	algoFilePackag	String		0	算法封装文件名称

序号	名称	标识符	类型	长度	必选/可选	说明
	名称	eName				
2	算法封装文件类型	algoFilePackageType	String		0	算法封装文件类型，BIN：二进制算法包，ENGINE：算法引擎
3	算法封装文件ID列表	ids	Long[]		0	算法封装文件ID列表
4	算法封装文件版本号	algoFilePackageVersion	String		0	算法封装文件版本号
5	算法提供方	provider	String		0	算法提供方
6	创建时间查询开始点	createTimeFrom	String		0	创建时间查询开始点 yyyy-MM-dd HH:mm:ss.sss， 例：2019-06-01 23:15:22.123
7	创建时间查询结束点	createTimeTo	String		0	创建时间查询结束点 yyyy-MM-dd HH:mm:ss.sss， 例：2019-06-01 23:15:22.123
8	边端设备编码	deviceCodes	String[]		0	如果边端设备编码列表不为空，则只查询已下发到这些边端设备的算法包

c) 响应参数：

表 A.16 AlgoFilePackageListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	数据	data	AlgoFilePackage[]		0	算法封装文件包列表
4	请求id	requestId	String		0	请求ID

表 A.17 AlgoFilePackage 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件名称	algoFilePackageName	String		R	算法封装文件名称

序号	名称	标识符	类型	长度	必选/可选	说明
2	算法封装文件类型	algoFilePackageType	String		R	算法封装文件类型，BIN：二进制算法包，ENGINE：算法引擎
3	算法封装文件版本号	algoFilePackageVersion	String		R	算法封装文件版本号
4	算法封装文件 ID	id	Long		R	算法封装文件 ID
5	算法引擎编码	engineCode	String		0	引擎类算法封装包导入后生成的算法引擎编码
6	算法提供方	provider	String		0	算法提供方
7	创建时间	createdDate	String		0	创建时间 yyyy-MM-dd HH:mm:ss.sss，例：2019-06-01 23:15:22.123

d) 请求示例：

```
{
  "id":1,
  "algoFilePackageName": "人体结构化",
  "algoFilePackageVersion": "1.0.0",
  "algoFilePackageType": "BIN",
  "provider": "baidu",
  "createTimeFrom": "2023-04-24 23:15:22",
  "createTimeTo": "2023-04-25 23:15:22",
  "deviceCodes": ["device1"],
}
```

e) 响应示例：

```
{
  "code": "0",
  "message": "success",
  "data": [
    {
      "code": "BAIDU-HUMAN-1.0.0",
      "algoFilePackageName": "人体结构化",
      "engineCode": "engine-1",
      "algoFilePackageVersion": "1.0.0",
      "algoFilePackageType": "BIN",
      "provider": "baidu",
      "createdDate": "2023-04-24 23:15:22"
    }
  ]
}
```

## A.5 查询平台算法封装文件详情

- a) URI: GET /ACP/algofilePackages/{id}  
 b) 请求参数:

表 A.18 请求参数信息

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件 ID	id	Long		R	算法封装文件 ID, 导入算法封装文件包后由平台生成

- c) 响应参数:

表 A.19 AlgoFilePackageDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	AlgoFilePackage		0	算法包信息
4	请求 id	requestId	String		0	请求 ID

- d) 请求示例:

GET/ACP/algofilePackages/1

- e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": {
    "code": 1,
    "algoFilePackageName": "人体结构化",
    "engineCode": "engine-1",
    "algoFilePackageVersion": "1.0.0",
    "algoFilePackageType": "BIN",
    "provider": "baidu",
    "createdDate": "2023-04-24 23:15:22"
  }
}
```

## A.6 创建算法能力

- a) URI: POST /ACP/algoDef  
b) 请求参数

表 A.20 TaskDefCreateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法名称	name	String		R	任务算法名称
2	任务算法编码	code	String		R	任务算法编码, 由算法提供方指定
3	任务算法参数类型	paramType	String		R	任务算法参数类型 (NORMAL: 普通参数, DEVICE: 设备参数 (只需要设备参数), AREA: 区域参数 (包含区域入参, 针对区域相关参数进行校验), AREA_LINE: 区域-有向线段参数 (包含区域和有向线段入参, 针对区域-有向线段相关参数进行校验), AREA_LINE_WITH_DIRECTION: 区域-线段参数 (包含区域和有向线段入参, 并支持直线, 针对区域-有向线段相关参数进行校验))
4	任务算法接入方式	sourceType	String		R	任务算法接入方式 (API: API, Engine: 引擎方式)
5	支持的数据源类型	taskTypes	String[]		R	VIDEO_REALTIME: 实时任务; VIDEO_PLAYBACK: 离线视频流任务; VIDEO_FILE: 离线视频文件任务; IMAGE_REALTIME: 图片流任务
6	算法源	source	String		R	API 地址或算法引擎编码, 算法引擎编码可通过 A.3、A4 查询接口返回的 engineCode 字段获取
7	任务算法描述	description	String		0	任务算法描述
8	任务算法数据类型	algoDataType	String		0	任务算法数据类型 (STRUCTURING: 结构化, ALERT: 告警, STATISTICS: 统计, OTHER: 其他)
9	任务算法参数列表	taskFields	TaskDefField[]		0	任务算法参数列表
10	回调数据字段描述	dataFields	TaskDefField[]		0	回调数据字段描述
12	任务算法版本	version	String		0	任务算法版本

表 A.21 TaskDefField 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	字段名称	name	String		R	任务算法字段名称
2	字段含义	text	String		R	任务算法字段含义
4	字段类型	type	String		R	任务算法字段类型, STRING: 字符串, BOOLEAN: 布尔类型, INTEGER: 整型, LONG: 长整型, DOUBLE: 浮点型, Image: base64 图片, Location: 位置
5	是否可见	visible	Boolean		R	任务算法字段是否可见
6	是否必填	required	Boolean		R	任务算法字段是否必填

7	字段描述	description	String		0	算法字段的描述信息
8	字段默认值	defaultValue	object		0	任务算法字段默认值
9	字段校验规则	validation	String		0	任务算法字段校验规则
10	字段分组	group	String		0	字段分组，可自定义字段分组，比如用于前端聚合展示或算法服务解析

## c) 响应参数

EmptyResponse

## d) 请求示例

```
{
  "name": "车辆跨线布控",
  "code": "VEHICLE_CROSS_LINE_ALERT",
  "description": "任务算法描述信息",
  "algoDataType": "STRUCTURING",
  "paramType": "NORMAL",
  "taskTypes": ["VIDEO_PLAYBACK", "VIDEO_REALTIME"],
  "fields": [
    {
      "name": "alertInterval",
      "text": "告警间隔",
      "description": "description",
      "type": "long",
      "visible": true,
      "required": true,
      "defaultValue": 10,
      "validation": "validation",
      "group": "group"
    }
  ],
  "sourceType": "ENGINE",
  "source": "algoCode",
  "version": ""
}
```

## e) 响应示例

```
{
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}
```

## A.7 更新算法能力

a) URI: PUT /ACP/algDef/{code}

b) 请求参数

表 A.22 TaskDefUpdateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法名称	name	String		R	任务算法名称
2	任务算法编码	code	String		R	任务算法编码
3	任务算法数据类型	algoDataType	String		R	任务算法数据类型（STRUCTURING：结构化，ALERT：告警，STATISTICS：统计，OTHER：其他）

4	任务算法接入方式	sourceType	String		R	任务算法接入方式 (API: API, ENGINE: 算法引擎)
5	算法源	source	String		R	API 地址或算法引擎编码, 算法引擎编码可通过 A.3、A4 查询接口返回的 engineCode 字段获取
6	任务算法参数类型	paramType	String		R	任务算法参数类型 (NORMAL: 普通参数, DEVICE: 设备参数 (只需要设备参数), AREA: 区域参数 (包含区域入参, 针对区域相关参数进行校验), AREA_LINE: 区域-有向线段参数 (包含区域和有向线段入参, 针对区域-有向线段相关参数进行校验), AREA_LINE_WITH_DIRECTION: 区域-线段参数 (包含区域和有向线段入参, 并支持直线, 针对区域-有向线段相关参数进行校验))
7	支持的数据源类型	taskTypes	String[]		R	VIDEO_REALTIME: 实时任务; VIDEO_PLAYBACK: 离线视频流任务; VIDEO_FILE: 离线视频文件任务; IMAGE_REALTIME: 图片流任务
8	任务算法参数列表	taskFields	TaskDefField[]		0	任务算法参数列表
9	回调数据字段描述	dataFields	TaskDefField[]		0	回调数据字段描述
10	任务算法描述	description	String		0	任务算法描述
11	任务算法版本	version	String		0	任务算法版本

## c) 响应参数

EmptyResponse

## d) 请求示例

PUT /ACP/algoDef/VEHICLE\_CROSS\_LINE\_ALERT

```
{
  "name": "车辆跨线布控",
  "code": "VEHICLE_CROSS_LINE_ALERT",
  "description": "任务算法描述信息",
  "algoDataType": "STRUCTURING",
  "paramType": "NORMAL",
  "taskTypes": ["VIDEO_PLAYBACK", "VIDEO_REALTIME"],
  "fields": [
    {
      "name": "alertInterval",
      "text": "告警间隔",
      "description": "description",
      "type": "long",
      "visible": true,
      "required": true,
      "advanced": true,
      "defaultValue": 30,
      "validation": "validation",
    }
  ]
}
```

```

    "group": "group"
  }
],
"sourceType": "ENGINE",
"source": "",
"version": "",
"id": 13
}

```

## e) 响应示例

```

{
  "data": {
    "id": 13
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

## A.8 A.7 查询算法能力列表

- a) URI: GET /ACP/algoDef  
b) 请求参数

表 A.23 TaskDefPageRequest 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法名称	name	String		0	任务算法名称
2	任务算法编码列表	taskDefCodes	String[]		0	任务算法编码列表
3	任务算法状态	status	String		0	任务算法状态 (UNTESTED: 未测试, APPLYING: 测试申请中, DEPLOYABLE: 可部署, DEPLOYING: 部署中, DEPLOY_FAILED: 部署失败, TESTABLE: 可测试, STOPPING: 停止中, PUBLISHABLE: 可发布, PUBLISHED: 已发布)
4	任务算法数据类型	algoDataType	String		0	任务算法数据类型 (STRUCTURING: 结构化, ALERT: 告警, STATISTICS: 统计, OTHER: 其他)
5	接入方式	sourceType	String		0	接入方式 (API: API, ENGINE: 算法引擎)
6	算法引擎编码列表	engineCodeList	String[]		0	算法引擎编码列表
7	分页参数	pageRequest	ApiPageRequest		0	
8	创建时间查询开始点	createTimeFrom	String		0	创建时间查询开始点
9	创建时间查询结束点	createTimeTo	String		0	创建时间查询结束点

表 A.24 ApiPageRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	页号	pageNumber	Integer		R	页号 [大小: 1~]
2	每页大小	pageSize	Integer		R	每页大小 [大小: 1~]
3	排序项	orderList	ApiOrder[]		0	排序项

表 A. 25 ApiOrder 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	排序属性	property	String		0	排序属性
2	排序方向	direction	String		0	排序方向

c) 响应参数

表 A. 26 TaskDefPageResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码(0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	TaskDefPageResponseData		0	分页结果数据
4	请求 id	requestId	String		0	请求 ID

表 A. 27 TaskDefPageResponseData 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	每页数据	pageContent	TaskDef[]		0	每页数据
2	总数	total	Integer		0	总数
3	分页条件	pageRequest	ApiPageRequest		0	分页条件

表 A. 28 TaskDef 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法名称	name	String		R	任务算法名称
2	任务算法编码	code	String		R	任务算法编码
3	算法提供方	provider	String		R	算法提供方
4	任务算法数据类型	algoDataType	String		R	任务算法数据类型 (STRUCTURING: 结构化, ALERT: 告警, STATISTICS: 统计, OTHER: 其他)
5	任务算法参数类型	paramType	String		R	任务算法参数类型 (NORMAL: 普通参数, DEVICE: 设备参数 (只需要设备参数), AREA: 区域参数 (包含区域-有向线段参数 (包含区域和有向线段入参, 针对区域-有向线段相关参数进行校验), AREA_LINE_WITH_DIRECTION: 区域-线段参数 (包含区域和有向线段入参, 并支持直线, 针相关参数进行校验))

6	接入方式	sourceType	String		R	接入方式 (API: API, ENGINE: 算法引擎)
7	算法引擎编码	source	String		R	API 地址或算法引擎编码, 算法引擎编码可通过 A.3、A4 查询接口返回的 engineCode 字段获取
8	支持的数据源类型	taskTypes	String[]		R	VIDEO_REALTIME: 实时任务; VIDEO_PLAYBACK: 离线视频流任务; VIDEO_FILE: 离线视频文件任务; IMAGE_REALTIME: 图片流任务
9	任务算法版本	version	String		0	任务算法版本
10	任务算法状态	status	String		0	任务算法状态 (UNTESTED: 未测试, APPLYING: 测试申请中, DEPLOYABLE: 可部署, DEPLOYING: 部署中, DEPLOY_FAILED: 部署失败, TESTABLE: 可测试, STOPPING: 停止中, PUBLISHABLE: 可发布, PUBLISHED: 已发布)
11	任务算法状态描述	statusDesc	String		0	任务算法状态描述
12	任务算法参数列表	taskFields	TaskDefField[]		0	任务算法参数列表
13	回调数据字段描述	dataFields	TaskDefField[]		0	回调数据字段描述
14	创建时间	createdDate	String		0	创建时间
15	修改时间	lastModifiedDate	String		0	修改时间
16	任务算法描述	description	String		0	任务算法描述

## d) 请求示例

```
{
  "name": "task name",
  "taskDefCodes": null,
  "status": "UNTESTED",
  "algoDataType": "STRUCTURING",
  "sourceType": "ENGINE",
  "engineCodeList": null,
  "taskType": "VIDEO_REALTIME",
  "pageRequest": {
    "pageNumber": 1,
    "pageSize": 20,
    "orderList": [
      {
        "property": "id",
        "direction": "desc"
      }
    ]
  },
  "createTimeFrom": "2020-01-01 10:00:00",
  "createTimeTo": "2030-01-01 23:00:00"
}
```

## e) 响应示例

```

{
  "data": {
    "pageRequest": {
      "pageNumber": 1,
      "pageSize": 20,
      "orderList": [
        {
          "property": "id",
          "direction": "desc"
        }
      ]
    },
    "pageContent": [
      {
        "name": "SFE 黄土裸露检测",
        "code": "SFE_BARE_LOESS_DETECT_ALERT",
        "description": "裸土大模型",
        "provider": "系统账户",
        "sourceType": "ENGINE",
        "source": "1",
        "status": "PUBLISHED",
        "taskTypes": ["VIDEO_PLAYBACK", "VIDEO_REALTIME"],
        "statusDesc": "已发布",
        "version": "0.1",
        "appliedQuota": 20,
        "quota": 100,
        "createdDate": "2020-01-01 10:00:00",
        "lastModifiedDate": "2020-01-01 10:00:00"
      }
    ],
    "total": 1
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

### A.9 查询算法能力详情

- a) URI: GET /ACP/algoDef/{code}
- b) 请求参数

表 A.29 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法编码	code	String		R	任务算法编码

- c) 响应参数

表 A.30 TaskDefDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	TaskDef		R	数据
4	请求 id	requestId	String		0	请求 ID

## d) 请求示例

GET /ACP/algoDef/BAIDU-HUMAN\_ATTR\_ALERT-1.0.0

## e) 响应示例

```
{
  "data": {
    "name": "",
    "code": "BAIDU-HUMAN_ATTR_ALERT-1.0.0",
    "description": "",
    "provider": "",
    "algoDataType": "STRUCTURING",
    "paramType": "",
    "sourceType": "",
    "source": "",
    "version": "",
    "status": "",
    "statusDesc": "",
    "taskTypes": ["VIDEO_PLAYBACK", "VIDEO_REALTIME"],
    "fields": [
      {
        "name": "alertInterval",
        "text": "告警间隔",
        "description": "description",
        "type": "long",
        "visible": true,
        "required": true,
        "advanced": true,
        "defaultValue": 30,
        "validation": "validation",
        "group": "group"
      }
    ]
  },
  "options": [
    {
      "name": "alertInterval",
      "text": "告警间隔",
      "description": "description",
      "type": "long",
      "visible": true,
      "required": true,
      "advanced": true,
      "defaultValue": 30,
      "validation": "validation",
      "group": "group"
    }
  ],
  "workloadCode": "",
  "createdDate": "2020-01-01 10:00:00",
}
```

```

    "lastModifiedDate": "2020-01-01 10:00:00"
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

#### A. 10删除算法能力

- a) URI: DELETE /ACP/algoDef/{code}
- b) 请求参数

表 A. 31 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务算法编码	code	String		R	任务算法编码

- c) 响应参数

表 A. 32 响应参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	请求 id	requestId	String		0	请求 ID

- d) 请求示例

DELETE /ACP/algoDef/BAIDU-HUMAN\_ATTR\_ALERT-1.0.0

- e) 响应示例

```

{
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

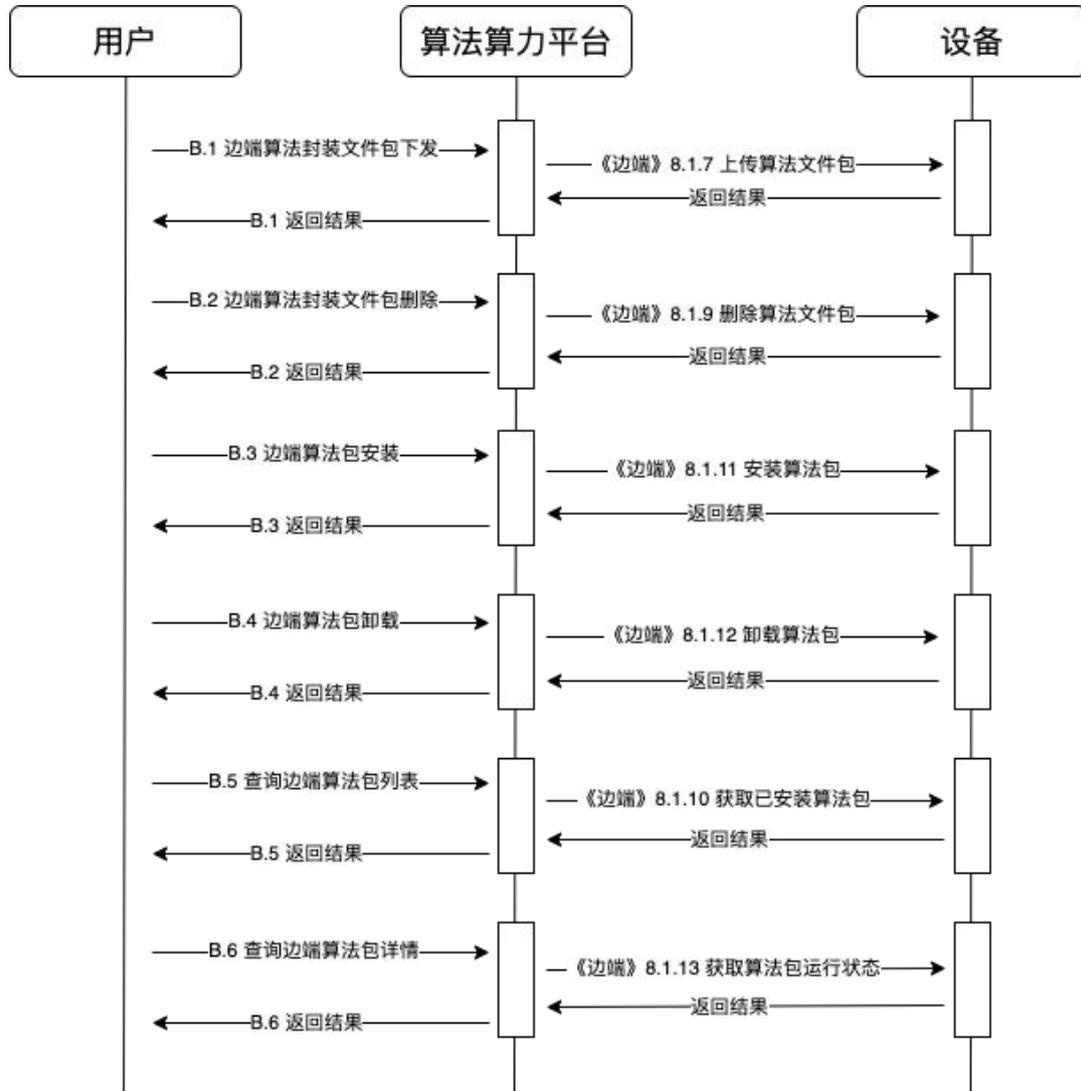
**附录 B**  
**(规范性)**  
**边端算法包部署**

**B.1 基本要求**

本章节共规定了6个边端算法包部署接口，主要包括算法封装文件包的下发、删除、安装、查询等，见表B.1。其中，算法算力平台与用户和设备之间的接口调度流程见图B.1，各个接口属性信息应符合表B.2-B.7的规定，具体应符合章节B.2-B.7的规定。

**表 B.1 边端算法包部署**

序号	URI	方法	功能说明
1	/ACP/algoFilePackage/devices	POST	下发边端算法封装文件包
2	/ACP/algoFilePackage/devices/{deviceCode}/{algoFilePackageId}	DELETE	删除边端算法封装文件包
3	/ACP/algoPackage/devices	POST	安装边端算法包
4	/ACP/algoPackage/devices/{deviceCode}/{algoPackageCode}	DELETE	卸载边端算法包
5	/ACP/algoPackage/devices	GET	查询边端算法包列表
6	/ACP/algoPackage/devices/{deviceCode}/{algoPackageCode}	GET	查询边端算法包详情



《视频图像智能感知前端和边缘设备适配技术要求》简称《边端》

图 B.1 边端算法包部署接口调度流程

表 B.2 下发边端算法封装文件包

URI	/ACP/algoFilePackage/devices		
功能	下发边端算法封装文件包		
方法	查询字符串	请求消息体	响应消息体
POST	无	AlgoFilePackageRequest	EmptyResponse
注释	将算法包下发到边端设备		

表 B.3 删除边端算法封装文件包

URI	/ACP/algoFilePackage/devices/{deviceCode}/{algoFilePackageId}		
功能	删除边端算法封装文件包		
方法	查询字符串	请求消息体	响应消息体
DELETE	无	无	EmptyResponse
注释	从边端设备删除算法包		

表 B.4 安装边端算法包

URI	/ACP/algoPackage/devices		
功能	安装边端算法包		
方法	查询字符串	请求消息体	响应消息体
POST	无	AlgoPackageInstallRequest	EmptyResponse
注释	边端算法包安装 算法包通过 A.1 算法封装文件包导入，算法封装文件包中应包含算法包编码 algoPackageCode，算法包编码由用户自定义，需要唯一，建议格式：provider-算法英文-version，由大小写字母、下划线、横线、英文.组成，长度小于128字符，例如 BAIDU-HUMAN_ATTR_ALERT-1.0.0。		

表 B.5 卸载边端算法包

URI	/ACP/algoPackage/devices/{deviceCode}/{algoPackageCode}		
功能	卸载边端算法包		
方法	查询字符串	请求消息体	响应消息体
DELETE	无	无	EmptyResponse
注释	边端算法包卸载		

表 B.6 查询边端算法包列表

URI	/ACP/algoPackage/devices		
功能	查询边端算法包列表		
方法	查询字符串	请求消息体	响应消息体
POST	无	AlgoPackageListRequest	AlgoPackageListResponse
注释	查询边端算法包列表		

表 B.7 查询边端算法包详情

URI	/ACP/algoPackage/devices/{deviceCode}/{algoPackageCode}		
功能	查询边端算法包详情		
方法	查询字符串	请求消息体	响应消息体
POST	无	无	AlgoPackageDetailResponse
注释	查询边端算法包详情		

## B.2 下发边端算法封装文件包

- a) URI: POST /ACP/algoFilePackage/devices
- b) 请求参数

表 B.8 AlgoFilePackageRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件包 ID	algoFilePackageId	Long		R	算法封装文件包 ID
2	边端设备编码	deviceCode	String		R	边端设备编码
3	鉴权地址	authAddress	String		0	鉴权地址，默认给中心地址 IP:AuthPort
4	鉴权文件地址	licenseFile	String		0	鉴权文件地址

- c) 响应参数  
EmptyResponse

d) 请求示例

```
{
  "algoFilePackageId":1,
  "authAddress":"127.0.0.1:8080",
  "licenseFile":"/auth_license.txt",
  "deviceCode":"device_code"
}
```

e) 响应示例

```
{
  "code": "0",
  "message": "success"
}
```

### B.3 删除边端算法封装文件包

a) URI: DELETE /ACP/algoFilePackage/{deviceCode}/{algoFilePackageId}

b) 请求参数

表 B.9 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法封装文件包 ID	algoFilePackageId	Long		R	算法封装文件包 ID
2	边端设备编码	deviceCode	String		R	边端设备编码

c) 响应参数

EmptyResponse

d) 请求示例

DELETE /ACP/algoFilePackage/device-1/1

e) 响应示例

```
{
  "code": "0",
  "message": "success"
}
```

### B.4 安装边端算法包

a) URI: POST /ACP/algoPackage/devices

b) 请求参数:

表 B.10 AlgoPackageInstallRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法包编码	algoPackageCode	String		R	算法包编码,用户在算法封装文件包中自定义
2	边端设备编码	deviceCode	String		R	边端设备编码
3	运行参数	args	String		0	运行参数

c) 响应参数

EmptyResponse

d) 请求示例

```
{
  "algoPackageCode": "BAIDU-HUMAN_ATTR_ALERT-1.0.0",
  "deviceCode": "device_code",
  "args": "xpuType=NVIDIA_T4"
}
```

e) 响应示例

```
{
  "code": "0",
  "message": "success"
}
```

## B.5 卸载边端算法包

a) URI: DELETE /ACP/algoPackage/devices/{deviceCode}/algoPackageCode

b) 请求参数:

表 B.11 请求参数属性

序号	名称	标识符	长度	必选/可选	说明
1	算法包编码	algoPackageCode		R	算法包编码
2	边端设备编码	deviceCode		R	边端设备编码

c) 响应参数

EmptyResponse

d) 请求示例

DELETE /ACP/algoPackage/devices/device-1/BAIDU-HUMAN\_ATTR\_ALERT-1.0.0

e) 响应示例

```
{
  "code": "0",
  "message": "success"
}
```

## B.6 查询边端算法包列表

a) URI: GET /ACP/algoPackage/devices

b) 请求参数

表 B.12 AlgoPackageListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	边端设备编码	deviceCode	String		R	边端设备编码
2	算法包运行状态	status	String		0	算法包运行状态(RUNNING: 运行中(安装即运行), STOPPED (停止)), 不传表示查全部状态的包

c) 响应参数:

表 B.13 AlgoPackageListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	数据	data	DeviceAlgoPackage[]		R	算法包列表
4	请求 id	requestId	String		0	请求 ID

表 B. 14 DeviceAlgoPackage 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法包名称	algoPackageName	String		R	算法包名称
2	算法包类型	algoPackageType	String		R	算法包类型，BIN：二进制算法包，ENGINE：算法引擎
3	算法包版本号	algoPackageVersion	String		R	算法包版本号
4	算法包编码	algoPackageCode	String		R	算法包编码
5	算法包运行状态	status	String		0	算法包运行状态（RUNNING：运行中，STOPPED（停止））
6	创建时间	createdDate	String		0	创建时间 yyyy-MM-dd HH:mm:ss.sss，例：2019-06-01 23:15:22.123

d) 请求示例：

```
{
  "deviceCode": "device_code",
  "status": "RUNNING"
}
```

e) 响应示例

```
{
  "code": "0",
  "message": "success",
  "data": [
    {
      "algoPackageCode": "BAIDU-HUMAN_ATTR_ALERT-1.0.0",
      "algoPackageName": "人体结构化",

```

```

    "algoPackageVersion": "1.0.0",
    "algoPackageType": "BIN",
    "status": "RUNNING",
    "createdDate": "2023-04-24 23:15:22"
  }
]
}

```

### B.7 查询边端算法包详情

- a) URI: GET /ACP/algopackage/devices/{deviceCode}/{algopackageCode}  
 b) 请求参数

表 B.15 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法包编码	algopackageCode	String		R	算法包编码
2	边端设备编码	deviceCode	String		R	边端设备编码

- c) 响应参数:

表 B.16 AlgoPackageDetailResponse

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	DeviceAlgoPackage		R	算法包信息
4	请求 id	requestId	String		0	请求 ID

- d) 请求示例:

GET /ACP/algopackage/devices/device-1/BAIDU-HUMAN\_ATTR\_ALERT-1.0.0

- e) 响应示例:

```

{
  "code": "0",
  "message": "success",
  "data": {
    "algopackageCode": "BAIDU-HUMAN_ATTR_ALERT-1.0.0",
    "algopackageName": "人体属性事件",
    "algoPackageVersion": "1.0.0",
    "algoPackageType": "BIN",
    "status": "RUNNING",
    "createdDate": "2023-04-24 23:15:22"
  }
}

```

T/BAX XXXX.5—2023

}  
}

**附录 C**  
**(规范性)**  
**算法算力服务管理平台任务调度**

**C.1 基本要求**

本章节共规定了14个算法算力平台任务调度接口，主要包括任务的创建、更新、查询和停止等，见表C.1。其中，算法算力平台与用户和中心/边端算力之间的接口调度流程见图C.1，各个接口属性信息应符合表C.2-C.15的规定，具体应符合章节C.2-C.15的规定。

**表 C.1 算法算力服务管理平台任务调度接口**

序号	URI	方法	功能说明
1	/ACP/tasks	POST	创建任务计划
2	/ACP/tasks/{id}	PUT	更新任务计划
3	/ACP/tasks	GET	查询任务计划列表
4	/ACP/tasks/{id}	GET	查询任务计划详情
5	/ACP/tasks/{id}	DELETE	删除任务计划
6	/ACP/tasks/start	POST	启动任务计划
7	/ACP/tasks/stop	POST	停止任务计划
8	/ACP/taskInstances	POST	添加任务实例到任务计划
9	/ACP/taskInstances/{id}	PUT	更新任务实例
10	/ACP/taskInstances	GET	查询任务实例列表
11	/ACP/taskInstances/{id}	GET	查询任务实例详情
12	/ACP/taskInstances/{id}	DELETE	删除任务实例
13	/ACP/taskInstances/start	POST	启动任务实例
14	/ACP/taskInstances/stop	POST	停止任务实例

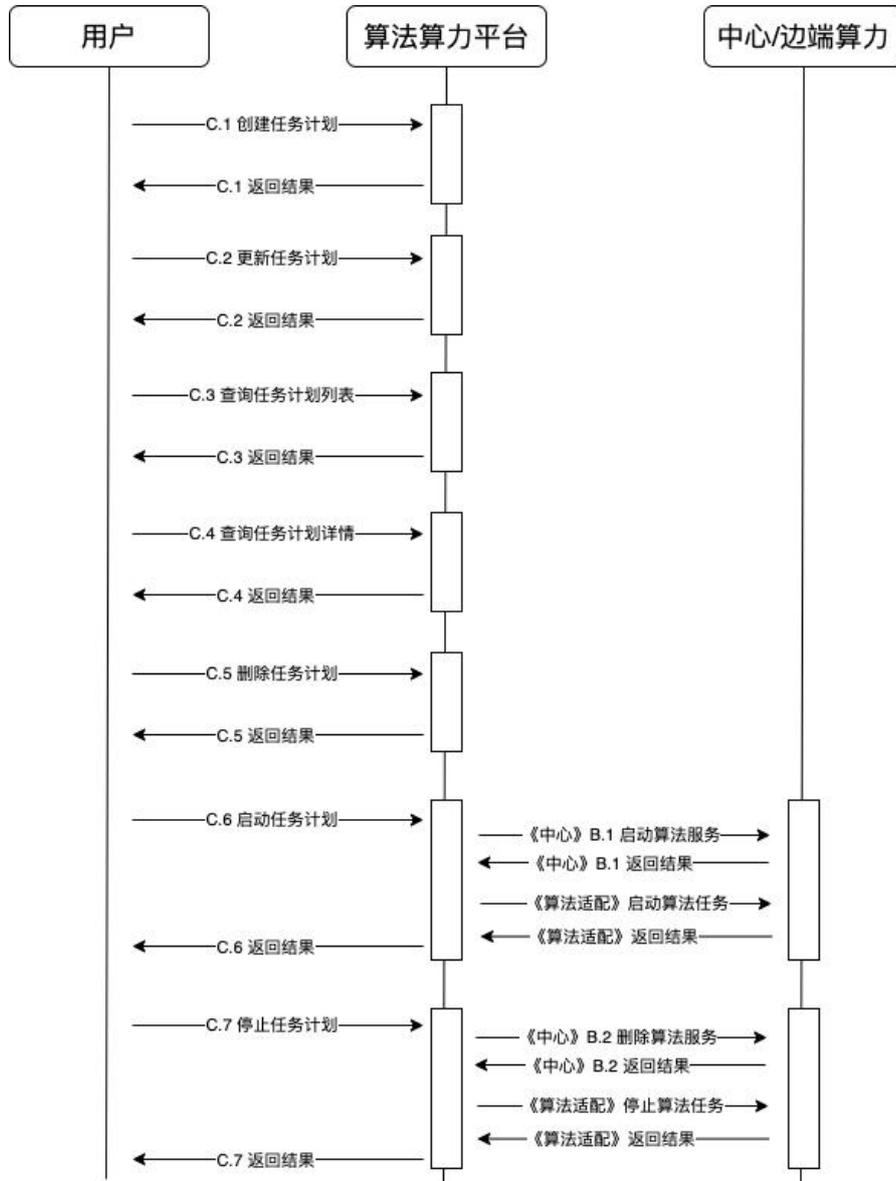


图 C.1 任务调度接口流程

表 C.2 创建任务计划接口

URI	/ACP/tasks		
功能	在算法算力服务管理平台创建任务计划		
方法	查询字符串	请求消息体	响应消息体
POST	无	TaskCreateRequest	TaskCreateResponse
注释	任务计划分可通过增加 playDayTime 开启周期启停，任务计划可设置巡检模式，巡检模式可按指定的并发数和单次时长轮询批量任务实例。巡检任务通过 inspectionPlan 参数设置巡检任务实例，非巡检任务通过 taskInstances 添加任务实例。		

表 C.3 更新任务计划

URI	/ACP/tasks/		
功能	更新任务计划		
方法	查询字符串	请求消息体	响应消息体

PUT	id	TaskUpdateRequest	TaskUpdateResponse
注释	在算法算力服务管理平台更新任务计划		

表 C.4 查询任务计划列表

URI	/ACP/tasks/		
功能	查询任务计划列表		
方法	查询字符串	请求消息体	响应消息体
GET	无	TaskListRequest	TaskListResponse
注释	在算法算力服务管理平台查询任务计划列表		

表 C.5 查询任务计划详情

URI	/ACP/tasks/		
功能	查询任务计划详情		
方法	查询字符串	请求消息体	响应消息体
POST	id	无	TaskDetailResponse
注释	在算法算力服务管理平台查询任务计划详情		

表 C.6 删除任务计划

URI	/ACP/tasks/		
功能	删除任务计划		
方法	查询字符串	请求消息体	响应消息体
DELETE	id	无	TaskDeleteResponse
注释	在算法算力服务管理平台删除任务计划详情		

表 C.7 开启任务计划

URI	/ACP/tasks/start		
功能	启动任务计划		
方法	查询字符串	请求消息体	响应消息体
POST	无		
注释	在算法算力服务管理平台启动任务计划		

表 C.8 停止任务计划

URI	/ACP/tasks/stop		
功能	停止任务计划		
方法	查询字符串	请求消息体	响应消息体
POST	无		
注释	在算法算力服务管理平台停止任务计划		

表 C.9 添加任务实例到任务计划

URI	/ACP/taskInstances		
功能	添加任务实例到任务计划		
方法	查询字符串	请求消息体	响应消息体
POST	无		
注释	在算法算力服务管理平台向任务计划添加实例		

表 C.10 更新任务实例

URI	/ACP/taskInstances		
功能	向任务计划添加实例		

方法	查询字符串	请求消息体	响应消息体
PUT	无		
注释	在算法算力服务管理平台向任务计划添加实例		

表 C.11 查询任务实例列表

URI	/ACP/taskInstances/		
功能	查询任务实例列表		
方法	查询字符串	请求消息体	响应消息体
GET		TaskInstanceListRequest	TaskInstanceListResponse
注释	在算法算力服务管理平台查询实例列表		

表 C.12 查询任务实例详情

URI	/ACP/taskInstances/{id}		
功能	查询任务实例详情		
方法	查询字符串	请求消息体	响应消息体
GET	id		TaskInstanceDetailResponse
注释	在算法算力服务管理平台查询任务实例详情		

表 C.13 删除任务实例

URI	/ACP/taskInstances/		
功能	删除任务实例		
方法	查询字符串	请求消息体	响应消息体
DELETE		TaskInstanceDeleteRequest	EmptyResponse
注释	在算法算力服务管理平台删除任务实例		

表 C.14 启动任务实例

URI	/ACP/taskInstances/start		
功能	启动任务实例		
方法	查询字符串	请求消息体	响应消息体
POST	无	TaskInstanceStartRequest	TaskInstanceStartResponse
注释	在算法算力服务管理平台启动任务实例		

表 C.15 停止任务实例

URI	/ACP/taskInstances/stop		
功能	停止任务实例		
方法	查询字符串	请求消息体	响应消息体
POST	无	TaskInstanceStopRequest	TaskInstanceStopResponse
注释	在算法算力服务管理平台停止任务实例		

## C.2 创建任务计划

- a) URI: POST /ACP/tasks  
 b) 请求参数:

表 C.16 TaskCreateRequest 属性

序号	名称	标识符	类型	长度	必选/ 可选	说明
1	任务计划名称	name	String		R	任务计划名称

2	任务计划类型	type	String		R	任务计划类型 (VIDEO_REALTIME: 实时视频流任务, VIDEO_PLAYBACK: 离线视频流任务, VIDEO_FILE: 离线视频文件任务, IMAGE_REALTIME: 图片流)
3	任务实例列表	taskInstances	TaskInstance[]		R	任务实例列表, 不 inspectionPlan 巡检参数时必填
4	任务计划描述	description	String		0	任务计划描述
5	巡检任务计划参数	inspectionPlan	TaskInspectionPlanCreateRequest		0	巡检任务计划参数, 传入巡检参数后使用巡检参数中的任务实例列表
6	任务计划生效时间段	planDayTime	String[]		0	任务计划生效时间段数组。每个元素英文逗号分割星期 (1-7)、起、止时间, 比如 "1, 7:00, 9:00" 表示"星期一7点到9点" 例如["1, 7:00, 9:00", "7, 12:00, 21:00"]
7	是否自动启动	autostart	Boolean		0	是否自动启动, 默认 false
8	是否自动重启	autoRestart	Boolean		0	任务失败时是否自动重启, 默认 true
9	离线任务计划请求参数	playbackTask	PlaybackTaskCreateRequest		0	离线任务计划请求参数, type=VIDEO_PLAYBACK/VIDEO_FILE 时必填

表 C.17 TaskInspectionPlanCreateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	巡检并发数	concurrency	Integer		R	巡检并发数 [大小: 1~]
2	单并发解析时长	duration	Integer		R	单并发解析时长(单位 s) [大小: 10~]
3	巡检设备列表	devices	TaskInspectionDevice[]		R	巡检设备列表
4	巡检算法信息	taskInstanceDef	TaskInspectionInstanceDef		R	巡检算法信息
5	巡检轮数限制	roundLimit	Integer		0	巡检轮数限制
6	数据回调地址	resultReceiveUrl	String[]		R	处理结果上传地址列表, 可以是自定义回调地址, 或满足 1400 规范, 例如 ["127.0.0.1:8080/receive", "10.0.0.1:8001/callback"]

表 C.18 TaskInspectionDevice 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	设备 ID	deviceId	String		R	设备 ID（国标编码）
2	区域列表	areas	RecDistrict[]		0	区域列表

表 C.19 TaskInspectionInstanceDef 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法编码	taskDefCode	String		R	算法编码
2	算法参数列表	taskFields	TaskInstanceField[]		0	算法参数列表
3	回调数据字段描述	dataFields	TaskInstanceField[]		0	回调数据字段描述

表 C.20 TaskInstance 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务实例 ID	taskInstanceId	Integer		R	任务实例 ID
2	算法编码	taskDefCode	String		R	算法编码
3	设备 ID	deviceId	String		R	设备 ID（国标编码）
4	数据回调地址	resultReceiveUrl	String[]		R	处理结果上传地址列表，可以是自定义回调地址，或满足 1400 规范，例如 ["127.0.0.1:8080/receive", "10.0.0.1:8001/callback"]
5	区域参数列表	areas	RecDistrict[]		0	任务实例区域参数列表
6	算法参数列表	taskFields	TaskInstanceField[]		0	算法参数列表
7	回调数据字段描述	dataFields	TaskInstanceField[]		0	回调数据字段描述
8	离线视频流任务开始时间	playbackStartTime	String		0	离线视频流任务开始时间
9	离线视频流任务结束时间	playbackEndTime	String		0	离线视频流任务结束时间

序号	名称	标识符	类型	长度	必选/可选	说明
	时间					
10	离线视频流任务当前解析时间	playbackCurrentTime	String		0	离线视频流任务当前解析时间
11	离线视频流任务抽帧倍速	speed	Integer		0	离线视频流任务抽帧倍速
12	视图库编码	viidCode	String		0	视图库编码
13	视图库订阅类别列表	viidSubscribeTypes	String[]		0	视图库订阅类别列表 (PERSON, FACE, MOTOR_VEHICLE, NON_MOTOR_VEHICLE, THING)
14	数据回调地址	resultReceiveUrl	String[]		R	处理结果上传地址列表，可以是自定义回调地址，或满足 1400 规范，例如 ["127.0.0.1:8080/receive", "10.0.0.1:8001/callback"]
15	优先级	priority	Integer	5	0	任务实例优先级，1-10，数字越大，优先级越高

表 C.21 RecDistrict 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	区域 ID	districtNo	Long		R	区域 ID
2	区域名称	districtName	String		R	区域名称
3	区域坐标	areaPoints	Point[]		R	区域相对坐标（可以是多边形）
4	有向线段坐标	linePoints	Point[]		0	有向线段相对坐标（eg: 0.25, 0.5, 0.75, 0.5）
5	方向	direction	String		0	方向（ALL: 双方向, CLOCKWISE: 顺时针, COUNTER_CLOCKWISE: 逆时针）

表 C.22 Point 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	x 坐标	pointX	Double		R	x 相对坐标
2	y 坐标	pointY	Double		R	y 相对坐标

表 C. 23 TaskInstanceField 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	参数名称	name	String		R	参数名称
2	参数值	value	object		R	参数值，类型任务算法字段类型，STRING：字符串，BOOLEAN：布尔类型，INTEGER：整型，LONG：长整型，DOUBLE：浮点型，Image：base64 图片，Location：位置
3	字段分组	group	String		0	字段分组

表 C. 24 PlaybackTaskCreateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	离线视频分片数	slice	Integer		R	离线视频分片数
2	离线视频流倍速	speed	Integer		R	离线视频流倍速 1/2/4/8/16/32
3	离线视频流任务开始时间	playbackStartTime	String		R	离线视频流任务开始时间
4	离线视频流任务结束时间	playbackEndTime	String		R	离线视频流任务结束时间

c) 响应参数：

表 C. 25 TaskCreateResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	数据	data	TaskStatus		R	任务计划状态描述
4	请求 id	requestId	String		0	请求 ID

表 C. 26 TaskStatus 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务计划 ID	id	Integer		0	taskId
2	任务状态	status	String		0	任务状态 (ON: 开启, OFF: 关闭)
3	任务实例状态描述	taskInstances	TaskInstanceStatus[]		0	任务实例状态描述

表 C.27 TaskInstanceStatus 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务实例 ID	id	Integer		0	任务实例 ID
2	任务实例状态	status	String		0	任务实例状态 (STARTING: 启动中, RUNNING: 运行中, STOPPING: 停止中, STOPPED: 已停止, START_ERROR: 启动异常, RUNNING_ERROR: 运行异常, WAITING: 等待中, RESTARTING: 重启中, EXCEPTION: 异常终止, PAUSED: 已暂停, FINISHED: 已完成)
3	任务启停标识	onOff	String		0	任务启停标识 (ON: 开启, OFF: 关闭)

## d) 请求示例

```
{
  "name": "task name",
  "type": "VIDEO_REALTIME",
  "description": "description for task",
  "inspectionPlan": {
    "concurrency": 10,
    "duration": 100,
    "resultReceiveUrl": ["localhost:8806/callback"],
    "devices": [
      {
        "deviceId": "uniqueDeviceId",
        "areas": [
          {
            "id": 1,
            "areaName": "area",
            "areaPoints": "0,0,1,0,1,1,0,1",
            "linePoints": "0.25,0.5,0.75,0.5",
            "direction": "ALL",
            "group": 1
          }
        ]
      }
    ]
  }
}
```

```

    }
  ],
  "taskInstanceDef": {
    "taskDefCode": "VEHICLE_CROSS_LINE_ALERT",
    "taskFields": [
      {
        "group": "group",
        "name": "alertInterval",
        "value": 10
      }
    ]
  },
  "dataFields": [
    {
      "group": "group",
      "name": "alertInterval",
      "value": 10
    }
  ]
},
"roundLimit": 1
},
"taskInstances": [
  {
    "taskInstanceId": 1,
    "resultReceiveUrl": ["localhost:8806/callback"],
    "priority": 5,
    "taskDefCode": "VEHICLE_CROSS_LINE_ALERT",
    "deviceId": "uniqueDeviceId",
    "areas": [
      {
        "id": 1,
        "areaName": "area",
        "areaPoints": "0,0,1,0,1,1,0,1",
        "linePoints": "0.25,0.5,0.75,0.5",
        "direction": "ALL",
        "group": 1
      }
    ]
  },
  {
    "taskFields": [
      {
        "group": "group",
        "name": "alertInterval",
        "value": 10
      }
    ]
  },
  {
    "dataFields": [
      {

```

```

        "group": "group",
        "name": "alertInterval",
        "value": 30
    }
],
    "playbackStartTime": "2022-11-11 12:15:22",
    "playbackEndTime": "2022-11-11 12:25:22",
    "playbackCurrentTime": "2022-11-11 12:25:22",
    "speed": 4,
    "viidCode": "32040200011327500001",
    "viidSubscribeTypes": null
}
],
"playDayTime": ["1,7:00-9:00", "7, 12:00, 21:00"],
"autostart": false,
"autoRestart": true,
"playbackTask": {
    "slice": 4,
    "speed": 4,
    "playbackStartTime": "2022-11-11 12:15:22",
    "playbackEndTime": "2022-11-11 12:25:22"
}
}
}
e) 响应示例
{
    "data": {
        "id": 1,
        "status": "ON",
        "taskInstances": [
            {
                "id": 1,
                "status": "RUNNING",
                "onOff": "ON"
            }
        ]
    },
    "code": "0",
    "message": "success",
    "requestId": "xxxx-xxxx-xxxx"
}

```

### C.3 更新任务计划

- a) URI: PUT /ACP/tasks/{id}
- b) 请求参数:

表 C.28 TaskUpdateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务计划ID	id	Integer		R	taskId
2	任务计划名称	name	String		R	任务计划名称
3	任务计划描述	description	String		0	任务计划描述
4	巡检任务计划参数	inspectionPlan	TaskInspectionPlanUpdateRequest		0	巡检任务计划参数
5	任务计划生效时间段	planDayTime	String[]		0	任务计划生效时间段数组。每个元素英文逗号分割星期(1-7)、起、止时间,比如“1,7:00,9:00”表示“星期一7点到9点” 例如[ “1,7:00,9:00”, “7,12:00-21:00” ]
6	是否自动重启	autoRestart	Boolean		0	任务失败时是否自动重启

表 C. 29 TaskInspectionPlanUpdateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	巡检并发数	concurrency	Integer		R	巡检并发数 [大小: 1~]
2	单并发解析时长	duration	Integer		R	单并发解析时长(单位 s) [大小: 10~]
3	数据回调地址	resultReceiveUrl	String[]		R	处理结果上传地址列表,可以是自定义回调地址,或满足 1400 规范,例如 [“127.0.0.1:8080/receive”,“10.0.0.1:8001/callback”]
4	巡检设备列表	devices	TaskInspectionDevice[]		0	巡检设备列表
5	巡检算法信息	taskInstanceDef	TaskInspectionInstanceDef		0	巡检算法信息

序号	名称	标识符	类型	长度	必选/可选	说明
6	巡检轮数限制	roundLimit	Integer		0	巡检轮数限制

c) 响应参数:

表 C.30 TaskUpdateResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	ID		R	ID
4	请求 id	requestId	String		0	请求 ID

表 C.31 ID 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	编号	id	Integer		R	taskId

d) 请求示例

```
{
  "id": 1,
  "name": "task name",
  "description": "description for task",
  "inspectionPlan": {
    "resultReceiveUrl": ["localhost:8806/callback"],
    "concurrency": 10,
    "duration": 100,
    "devices": [
      {
        "deviceId": "uniqueDeviceId",
        "areas": [
          {
            "id": 1,
            "areaName": "area",
            "areaPoints": "0,0,1,0,1,1,0,1",
            "linePoints": "0.25,0.5,0.75,0.5",
            "direction": "ALL",
            "group": 1
          }
        ]
      }
    ]
  }
}
```

```

    ],
    "taskInstanceDef": {
      "taskDefCode": "VEHICLE_CROSS_LINE_ALERT",
      "fields": [
        {
          "group": "group",
          "name": "alertInterval",
          "value": {}
        }
      ],
      "options": [
        {
          "group": "group",
          "name": "alertInterval",
          "value": {}
        }
      ]
    },
    "roundLimit": 1
  },
  "playDayTime": ["1, 7:00-9:00", "7, 12:00, 21:00"],
  "autoRestart": true
}

```

## e) 响应示例

```

{
  "data": {
    "id": 13
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

## C.4 查询任务计划列表

- a) URI: GET /ACP/tasks  
b) 请求参数:

表 C.32 TaskListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务计划 ID 列表	taskIds	Integer[]		0	taskId 列表
2	任务计划状态	status	String		0	任务计划状态 (ON: 开启, OFF: 关闭)
3	任务计划名称	name	String		0	任务计划名称
4	任务计划类型	type	String		0	任务计划类型 (VIDEO_REALTIME: 实时视频流任务, VIDEO_PLAYBACK: 离线视频流任务, VIDEO_FILE: 离线视频文件任务, IMAGE_REALTIME: 图片)

5	任务计划执行模式	mode	String		0	任务计划执行模式 (PERIODIC: 周期执行模式, INSPECTIVE: 巡检执行模式)
6	算法名称	taskDefName	String		0	算法名称
7	设备 ID 列表	deviceIds	String[]		0	设备 ID 列表
8	算法编码列表	taskDefCodes	String[]		0	算法编码列表
9	分页参数	pageRequest	ApiPageRequest		0	默认值 pageNumber=1, pageSize=10
10	创建时间查询开始点	createTimeFrom	String		0	创建时间查询开始点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22
11	创建时间查询结束点	createTimeTo	String		0	创建时间查询结束点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22

c) 响应参数:

表 C. 33 TaskListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	TaskPageResponse		R	分页结果数据
4	请求 id	requestId	String		0	请求 ID

表 C. 34 TaskPageResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	每页数据	pageContent	TaskListItem[]		0	每页数据
2	总数	total	Integer		0	总数
3	分页条件	pageRequest	ApiPageRequest		0	分页条件

表 C. 35 TaskListItem 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	编号	id	Integer		R	taskId
2	任务计划名称	name	String		0	任务计划名称
3	任务计划描述	description	String		0	任务计划描述

4	任务计划状态	status	String		0	任务计划状态 (ON: 开启, OFF: 关闭)
5	任务计划类型	type	String		0	任务计划类型 (VIDEO_REALTIME: 实时视频流任务, VIDEO_PLAYBACK: 离线视频流任务, VIDEO_FILE: 离线视频文件任务, IMAGE_REALTIME: 图片)
6	任务计划执行模式	mode	String		0	任务计划执行模式 (PERIODIC: 周期执行模式, INSPECTIVE: 巡检执行模式)
7	巡检任务计划信息	inspectionPlan	TaskInspectionPlanListItem		0	巡检任务计划信息
8	设备数	deviceCount	Integer		0	设备数
9	任务实例总数	totalCount	Integer		0	任务实例总数
10	启动中实例数	startingCount	Integer		0	启动中实例数
11	启动异常实例数	startErrCount	Integer		0	启动异常实例数
12	运行中实例数	runningCount	Integer		0	运行中实例数
13	关闭实例数	stoppedCount	Integer		0	关闭实例数
14	已完成实例数	finishedCount	Integer		0	已完成实例数
15	任务计划生效日	planDay	String		R	任务计划生效日
16	任务计划生效时间段	planTime	String		R	任务计划生效时间段
17	离线任务列表项	playbackTaskListItem	PlaybackTaskListItem		0	离线任务列表项, 仅离线任务计划有此字段
18	创建时间	createdDate	String		0	创建时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123
19	修改时间	lastModifiedDate	String		0	修改时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123

表 C. 36 TaskInspectionPlanListItem 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	巡检并发数	concurrency	Integer		0	巡检并发数
2	单并发解析时长	duration	Integer		0	单并发解析时长(单位 s)
3	巡检轮次计数	roundCount	Integer		0	巡检轮次计数
4	当前轮巡检进度	progress	Integer		0	当前轮巡检进度
5	巡检轮数限制	oneOffRoundLimit	Integer		0	巡检轮数限制
6	是否自动重启	autoRestart	Boolean		0	任务失败时是否自动重启

表 C.37 PlaybackTaskListItem 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	离线视频分片数	slice	Integer		0	离线视频分片数
2	整体加速倍数	totalAcc	Integer		0	整体加速倍数(分片数*抽帧倍速)

d) 请求示例

```
{
  "taskIds": [
    1
  ],
  "status": "ON",
  "name": "task name",
  "type": "VIDEO_REALTIME",
  "mode": "PERIODIC",
  "taskDefName": "车辆跨线布控",
  "deviceIds": null,
  "taskDefCodes": null,
  "pageRequest": {
    "pageNumber": 1,
    "pageSize": 20,
    "orderList": [
      {
        "property": "id",
        "direction": "desc"
      }
    ]
  }
},
"createTimeFrom": "2020-01-01 10:00:00",
```

```

    "createTimeTo": "2030-01-01 23:00:00"
  }
e) 响应示例
{
  "data": {
    "pageRequest": {
      "pageNumber": 1,
      "pageSize": 20,
      "orderList": [
        {
          "property": "id",
          "direction": "desc"
        }
      ]
    },
    "pageContent": [
      {
        "id": 13,
        "name": "人流量布控任务",
        "autoRestart": true,
        "description": "任务计划描述",
        "status": "ON",
        "type": "VIDEO_REALTIME",
        "mode": "PERIODIC",
        "inspectionPlan": {
          "resultReceiveUrl": ["localhost:8806/callback"],
          "concurrency": 50,
          "duration": 60,
          "roundCount": 20,
          "progress": 20,
          "oneOffRoundLimit": 1
        },
        "deviceCount": 200,
        "totalCount": 100,
        "startingCount": 5,
        "startErrCount": 4,
        "runningCount": 20,
        "stoppedCount": 3,
        "finishedCount": 5,
        "playDayTime": ["1, 7:00-9:00", "7, 12:00, 21:00"],
        "playbackTaskListItem": {
          "slice": 4,
          "totalAcc": 16
        },
        "createdDate": "2020-01-01 10:00:00",
        "lastModifiedDate": "2020-01-01 10:00:00"
      }
    ]
  }
}

```

```

    ],
    "total": 1
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

### C.5 查询任务计划详情

- a) URI: GET /ACP/tasks/{id}  
 b) 请求参数

表 C.38 请求参数:属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	编号	id	Integer		R	taskId

- c) 响应参数

表 C.39 TaskDetailResponse 属性

序号	名称	标识符	类型	长度	必选	说明
1	结果状态码	code	String		R	结果状态码(0:成功,其他:失败)
2	结果描述	message	String		R	结果描述
3	数据	data	TaskDetail		R	任务详情
4	请求 id	requestId	String		0	请求 ID

表 C.40 TaskDetail 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	任务实例列表	taskInstances	TaskInstanceDetail[]		0	任务实例列表
2	巡检任务计划	inspectionPlan	TaskInspectionPlanDetail		0	巡检任务计划
3	任务计划名称	name	String		R	任务计划名称
4	任务计划描述	description	String		0	任务计划描述
5	任务计划类型	type	String		0	任务计划类型 (VIDEO_REALTIME: 实时视频流任务, VIDEO_PLAYBACK: 离线视频流任务, VIDEO_FILE: 离线视频文件任务,

序号	名称	标识符	类型	长度	必选/可选	描述
						IMAGE_REALTIME: 图片)
6	任务计划执行模式	mode	String		0	任务计划执行模式 (PERIODIC: 周期执行模式, INSPECTIVE: 巡检执行模式)
7	任务计划状态	status	String		0	任务计划状态 (ON: 开启, OFF: 关闭)
8	巡检周期	roundTime	Integer		0	巡检周期 (单位: 秒)
9	任务计划生效时间段	planDayTime	String[]		R	任务计划生效时间段数组。每个元素英文逗号分割星期(1-7)、起、止时间, 比如“1, 7:00, 9:00”表示“星期一7点到9点” 例如[ “1, 7:00, 9:00”, “7, 12:00-21:00” ]
10	创建时间	createdDate	String		0	创建时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123
11	修改时间	lastModifiedDate	String		0	修改时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123
12	任务计划ID	id	Integer		0	taskId
13	是否自动重启	autoRestart	Boolean		0	任务失败时是否自动重启

表 C.41 TaskInstanceDetail 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	区域列表	areas	RecDistrict[]		0	
2	算法参数列表	taskFields	TaskInstanceField[]		0	
3	回调数据字段描述	dataFields	TaskInstanceField[]		0	
4	taskId	taskId	Integer		0	

序号	名称	标识符	类型	长度	必选/可选	描述
5	任务计划名称	taskName	String		0	
6	算法名称	taskDefName	String		0	
7	算法编码	taskDefCode	String		0	
8	设备 ID	deviceId	String		0	
9	设备名称	deviceName	String		0	
10	任务实例状态	status	String		0	
11	任务实例状态描述	statusDesc	String		0	
12	离线视频流任务开始时间	playbackStartTime	String		0	
13	离线视频流任务结束时间	playbackEndTime	String		0	
14	离线视频流任务当前解析时间	playbackCurrentTime	String		0	
15	离线任务当前解析进度百分比	playbackProgress	String		0	
16	离线任务时长描述	playbackLengthDesc	String		0	
17	离线任务已解析时长描述	completedLengthDesc	String		0	
18	离线视频流任务抽帧倍速	speed	Integer		0	
19	视图库编码	viidCode	String		0	
20	视图库订阅类别列表	viidSubscribeTypes	String[]		0	
21	创建时间	createdDate	String		0	
22	修改时间	lastModifiedDate	String		0	
23	任务实例 ID	id	Integer		0	

序号	名称	标识符	类型	长度	必选/可选	描述
24	回调地址	resultReceiveUrl	String[]		R	
25	优先级	priority	Integer		0	

表 C.42 TaskInspectionPlanDetail 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	巡检并发数	concurrency	Integer		0	巡检并发数
2	单并发解析时长	duration	Integer		0	单并发解析时长
3	巡检轮次计数	roundCount	Integer		0	巡检轮次计数
4	当前轮巡检进度	progress	Integer		0	当前轮巡检进度
5	巡检轮数限制	roundLimit	Integer		0	巡检轮数限制，完成指定轮次巡检后关闭巡检计划。
6	巡检设备	devices	TaskInspectionDevice		0	巡检设备
7	巡检算法信息	taskInstanceDef	TaskInspectionInstanceDef		0	巡检算法信息
8	数据回调地址	resultReceiveUrl	String[]		R	处理结果上传地址列表，可以是自定义回调地址，或满足 1400 规范，例如 ["127.0.0.1:8080/receive", "1

序号	名称	标识符	类型	长度	必选/可选	描述
						0.0.0.1:8001/callback"]

d) 请求示例

```
{
  "id": 1
}
```

e) 响应示例

```
{
  "data": {
    "taskInstances": [
      {
        "resultReceiveUrl": ["localhost:8806/callback"],
        "priority": 5,
        "areas": [
          {
            "id": 1,
            "areaName": "area",
            "areaPoints": "0,0,1,0,1,1,0,1",
            "linePoints": "0.25,0.5,0.75,0.5",
            "direction": "ALL",
            "group": 1
          }
        ]
      },
      {
        "taskFields": [
          {
            "group": "group",
            "name": "alertInterval",
            "value": 30
          }
        ]
      },
      {
        "dataFields": [
          {
            "group": "group",
            "name": "alertInterval",
            "value": 30
          }
        ]
      },
      {
        "taskId": 1,
        "taskName": "",
        "taskDefName": "",
        "taskDefCode": "",
        "deviceId": ""
      }
    ]
  }
}
```

```

        "deviceName": "",
        "status": "",
        "statusDesc": "",
        "playbackStartTime": "2022-11-11 12:15:22",
        "playbackEndTime": "2022-11-11 12:25:22",
        "playbackCurrentTime": "2022-11-11 12:20:22",
        "playbackProgress": "92.3",
        "playbackLengthDesc": "1 小时 12 分",
        "completedLengthDesc": "1 小时 2 分",
        "speed": 4,
        "compareTaskId": 1,
        "viidCode": "",
        "viidSubscribeTypes": null,
        "createdDate": "2020-01-01 10:00:00",
        "lastModifiedDate": "2020-01-01 10:00:00",
        "id": 13
    }
],
"inspectionPlan": {
    "resultReceiveUrl": ["localhost:8806/callback"],
    "concurrency": 1,
    "duration": 1,
    "roundCount": 1,
    "progress": 1,
    "roundLimit": 1,
    "devices": [
        {
            "deviceId": "uniqueDeviceId",
            "areas": [
                {
                    "id": 1,
                    "areaName": "area",
                    "areaPoints": "0,0,1,0,1,1,0,1",
                    "linePoints": "0.25,0.5,0.75,0.5",
                    "direction": "ALL",
                    "group": 1
                }
            ]
        }
    ]
}
],
"taskInstanceDef": {
    "taskDefCode": "VEHICLE_CROSS_LINE_ALERT",
    "taskFields": [
        {
            "group": "group",
            "name": "alertInterval",
            "value": 30
        }
    ]
}

```

```

    }
  ],
  "dataFields": [
    {
      "group": "group",
      "name": "alertInterval",
      "value": 30
    }
  ]
}
},
"name": "task name",
"autoRestart": true,
"description": "description for task",
"type": "VIDEO_REALTIME",
"mode": "PERIODIC",
"status": "ON",
"roundTime": 1,
"playDayTime": ["1, 7:00-9:00", "7, 12:00, 21:00"],
"createdDate": "2020-01-01 10:00:00",
"lastModifiedDate": "2020-01-01 10:00:00",
"id": 13
},
"code": "0",
"message": "success",
"requestId": "xxxx-xxxx-xxxx"
}

```

## C.6 删除任务计划

- a) URI: DELETE /ACP/tasks/{id}  
 b) 请求参数

表 C.43 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务计划 ID	id	Integer		R	taskId

- c) 响应参数:

表 C.44 TaskDeleteResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述

序号	名称	标识符	类型	长度	必选/可选	说明
3	数据	data	ID		R	任务计划 ID
4	请求 id	requestId	String		0	请求 ID

d) 请求示例

```
{
  "id": 1
}
```

e) 响应示例

```
{
  "data": {
    "id": 13
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}
```

### C.7 启动任务计划

a) URI: POST /ACP/tasks/start

b) 请求参数:

表 C.45 TaskStartRequest 属性

序号	名称	标识符	类型	长度	必选	说明
1	任务计划 ID	id	Integer		R	taskId

c) 响应参数

表 C.46 TaskStartResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码(0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	taskInstances		R	任务计划, 见 C.4 详细描述
4	请求 id	requestId	String		0	请求 ID

d) 请求示例

```
{
  "id": 1
}
```

e) 响应示例

```

{
  "data": {
    "id": 1,
    "status": "ON",
    "taskInstances": [
      {
        "id": 1,
        "status": "RUNNING",
        "onOff": "ON"
      }
    ]
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

### C.8 停止任务计划

a) URI: POST /ACP/tasks/stop

b) 请求参数:

表 C.47 TaskStopRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务计划 ID	id	Integer		R	taskId

c) 响应参数:

表 C.48 TaskStopResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	taskInstances		R	任务计划, 见 C.4 详细描述
4	请求 id	requestId	String		0	请求 ID

d) 请求示例

```

{
  "id": 1
}

```

e) 响应示例

```

{
  "data": {

```

```

    "id": 1,
    "status": "ON",
    "taskInstances": [
      {
        "id": 1,
        "status": "RUNNING",
        "onOff": "ON"
      }
    ]
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

**C.9 添加任务实例到任务计划**

- a) 路径: POST /ACP/taskInstances
- b) 请求参数:

**表 C.49 CreatetaskInstancesRequest 属性**

序号	名称	类型	长度	必选/可选	描述
1	taskId	Integer		R	taskId
2	taskInstances	TaskInstance[]		R	任务实例列表, 见 C.4 详细描述
3	autostart	Boolean		0	添加任务实例成功后 是否自动启动 (默认为 false)

- c) 响应参数:

**表 C.50 CreatetaskInstancesResponse 属性**

序号	名称	类型	长度	必选/可选	描述
1	data	TaskInstanceStatus[]		0	任务实例状态, 见 C.4 详细描述
2	code	String		0	状态码
3	message	String		0	状态消息
4	requestId	String		0	请求 ID

- d) 请求示例

```

{
  "taskId": 1,
  "taskInstances": [
    {
      "resultReceiveUrl": ["localhost:8806/callback"],
      "priority": 5,
    }
  ]
}

```

```

    "taskDefCode": "VEHICLE_CROSS_LINE_ALERT",
    "deviceId": "uniqueDeviceId",
    "areas": [
      {
        "id": 1,
        "areaName": "area",
        "areaPoints": "0,0,1,0,1,1,0,1",
        "linePoints": "0.25,0.5,0.75,0.5",
        "direction": "ALL",
        "group": 1
      }
    ],
    "fields": [
      {
        "group": "group",
        "name": "alertInterval",
        "value": {}
      }
    ],
    "options": [
      {
        "group": "group",
        "name": "alertInterval",
        "value": {}
      }
    ],
    "playbackStartTime": "2022-11-11 12:15:22",
    "playbackEndTime": "2022-11-11 12:25:22",
    "playbackCurrentTime": "2022-11-11 12:25:22",
    "speed": 4,
    "viidCode": "32040200011327500001",
    "viidSubscribeTypes": null
  }
]
}

```

## e) 响应示例

```

{
  "data": [
    {
      "id": 1,
      "status": "RUNNING",
      "onOff": "ON"
    }
  ],
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

}

**C. 10更新任务实例**

- a) 路径: PUT /ACP/taskInstances  
 b) 请求参数:

**表 C. 51 TaskInstanceUpdateRequest 属性**

序号	名称	标识符	类型	长度	必选/可选	描述
1	id	id	Integer		R	任务实例 ID
2	deviceId	deviceId	String		R	设备 ID
3	resultReceiveUrl	resultReceiveUrl	String[]		R	处理结果上传地址列表, 可以是自定义回调地址, 或满足 1400 规范, 例如 ["127.0.0.1:8080/receive", "10.0.0.1:8001/callback"]
4	areas	areas	RecDistrict[]		0	区域列表
5	taskFields	taskFields	TaskInstanceField[]		0	算法参数列表
6	dataFields	dataFields	TaskInstanceField[]		0	回调数据字段描述
7	priority	priority	Integer	5	0	任务实例优先级, 1-10, 数字越大, 优先级越高

- c) 响应参数:

**表 C. 52 TaskInstanceUpdateResponse 属性**

序号	名称	标识符	类型	长度	必选/可选	描述
1	data	data	ID		0	ID
3	code	code	String		0	状态码
4	message	message	String		0	状态消息
5	requestId	requestId	String		0	请求 ID

- d) 请求示例

```
{
  "id": 1,
  "deviceId": "uniqueDeviceId",
  "areas": [
    {
      "id": 1,
      "resultReceiveUrl": ["localhost:8806/callback"],
      "priority": 5,

```

```

    "areaName": "area",
    "areaPoints": "0,0,1,0,1,1,0,1",
    "linePoints": "0.25,0.5,0.75,0.5",
    "direction": "ALL",
    "group": 1
  }
],
"fields": [
  {
    "group": "group",
    "name": "alertInterval",
    "value": {}
  }
],
"options": [
  {
    "group": "group",
    "name": "alertInterval",
    "value": {}
  }
]
}

```

## e) 响应示例

```

{
  "data": {
    "id": 13
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

## C.11 查询任务实例列表

- a) 路径: GET /ACP/taskInstances  
b) 请求参数

表 C.53 TaskInstanceListRequest 属性

序号	名称	类型	长度	必选/可选	描述
1	id	Integer		0	任务实例 ID

序号	名称	类型	长度	必选/可选	描述
2	status	String		0	任务实例状态 (STARTING: 启动中, RUNNING: 运行中, STOPPING: 停止中, STOPPED: 已停止, START_ERROR: 启动异常, RUNNING_ERROR: 运行异常, WAITING: 等待中, RESTARTING: 重启中, EXCEPTION: 异常终止, PAUSED: 已暂停, FINISHED: 已完成)
3	taskDefCodes	String[]		0	任务算法编码列表
4	taskDefName	String		0	任务算法名称
5	taskId	Integer		0	任务 ID
6	taskName	String		0	任务名称
7	deviceIds	String[]		0	设备 ID 列表
8	pageRequest	ApiPageRequest		0	
9	createTimeFrom	String		0	创建时间查询开始点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22
10	createTimeTo	String		0	创建时间查询结束点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22
11	priority	Integer		0	任务实例优先级, 1-10, 数字越大, 优先级越高

## c) 响应参数

表 C. 54 TaskInstanceListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	每页数据	pageContent	TaskInstanceDetail[]		0	每页数据
2	总数	total	Integer		0	总数
3	分页条件	pageRequest	ApiPageRequest		0	分页条件

## d) 请求示例

```

{
  "id": 1,
  "status": "STOPPED",
  "taskDefCodes": null,
  "taskDefName": "车辆跨线布控",
  "taskId": 1,
  "taskName": "task name",
  "deviceIds": null,
  "pageRequest": {
    "pageNumber": 1,
    "pageSize": 20,
    "orderList": [
      {
        "property": "id",
        "direction": "desc"
      }
    ]
  },
  "createTimeFrom": "2020-01-01 10:00:00",
  "createTimeTo": "2030-01-01 23:00:00"
}

```

## e) 响应示例

```

{
  "data": {
    "pageRequest": {
      "pageNumber": 1,
      "pageSize": 20,
      "orderList": [
        {
          "property": "id",
          "direction": "desc"
        }
      ]
    },
    "pageContent": [
      {}
    ],
    "total": 1
  },
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}

```

## C.12 查询任务实例详情

- a) 路径: GET /ACP/taskInstances/{id}  
 b) 请求参数

表 C.55 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	任务计划 ID	id	Integer		R	任务计划 ID

- c) 响应参数

表 C.56 响应参数属性

序号	名称	类型	长度	必选/可选	描述
1	data	TaskInstanceDetail		0	
2	code	String		0	状态码
3	message	String		0	状态消息
4	requestId	String		0	请求 ID

- d) 请求示例

```
{
  "id": 1
}
```

- e) 响应示例

```
{
  "data": {
    "resultReceiveUrl": ["localhost:8806/callback"],
    "priority": 5,
    "areas": [
      {
        "id": 1,
        "areaName": "area",
        "areaPoints": "0,0,1,0,1,1,0,1",
        "linePoints": "0.25,0.5,0.75,0.5",
        "direction": "ALL",
        "group": 1
      }
    ],
    "fields": [
      {
        "group": "group",
        "name": "alertInterval",
        "value": {}
      }
    ],
    "options": [
      {
```

```

    "group": "group",
    "name": "alertInterval",
    "value": {}
  }
],
  "taskId": 1,
  "taskName": "",
  "taskDefName": "",
  "taskDefCode": "",
  "deviceId": "",
  "deviceName": "",
  "status": "",
  "statusDesc": "",
  "playbackStartTime": "2022-11-11 12:15:22",
  "playbackEndTime": "2022-11-11 12:25:22",
  "playbackCurrentTime": "2022-11-11 12:20:22",
  "playbackProgress": "92.3",
  "playbackLengthDesc": "1 小时 12 分",
  "completedLengthDesc": "1 小时 2 分",
  "speed": 4,
  "compareTaskId": 1,
  "viidCode": "",
  "viidSubscribeTypes": null,
  "createdDate": "2020-01-01 10:00:00",
  "lastModifiedDate": "2020-01-01 10:00:00",
  "id": 13
},
"code": "0",
"message": "success",
"requestId": "xxxx-xxxx-xxxx"
}

```

### C.13 删除任务实例

- a) URI: DELETE /ACP/taskInstances  
 b) 请求参数:

表 C.57 TaskInstanceDeleteRequest 属性

序号	名称	类型	长度	必选/可选	说明
1	ids	Integer[]		R	ID 列表

- c) 响应参数:

EmptyResponse

- d) 请求示例

```

{
  "ids": [
    1
  ]
}

```

e) 响应示例

```
{
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}
```

## C.14 启动任务实例

a) URI: POST /ACP/taskInstances/start

b) 请求参数:

表 C.58 TaskInstanceStartRequest 属性

序号	名称	类型	长度	必选/可选	说明
1	ids	Integer[]		R	ID 列表

c) 响应参数:

表 C.59 TaskInstanceStartResponse 属性

序号	名称	类型	长度	必选/可选	说明
1	data	TaskInstanceStatusListData		0	
5	code	String		0	状态码
6	message	String		0	状态消息
7	requestId	String		0	请求 ID

表 C.60 TaskInstanceStatusListData 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	taskInstanceStatusList	taskInstanceStatusList	TaskInstanceStatus[]		R	任务实例状态

表 C.61 TaskInstanceStatus 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	id	id	Integer		R	任务实例 ID
2	status	status	String		R	任务实例状态 (STARTING: 启动中, RUNNING: 运行中, STOPPING: 停止中, STOPPED: 已停止)

序号	名称	标识符	类型	长度	必选/可选	描述
						已停止, START_ERROR: 启动异常, RUNNING_ERROR: 运行异常, WAITING: 等待中, RESTARTING: 重启中, EXCEPTION: 异常终止, PAUSED: 已暂停, FINISHED: 已完成)
3	onOff	onOff	String		R	任务启停标识 (ON: 开启, OFF: 关闭)

d) 请求示例

```
{
  "ids": [
    1
  ]
}
```

e) 响应示例

```
{
  "data": [
    {
      "id": 1,
      "status": "RUNNING",
      "onOff": "ON"
    }
  ],
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}
```

### C.15 停止任务实例

a) URI: POST /ACP/taskInstances/stop

b) 请求参数:

表 C.62 TaskInstanceStopRequest 属性

序号	名称	类型	长度	必选/可选	说明
1	ids	Integer[]		R	ID 列表

c) 响应参数:

表 C.63 TaskInstanceStopResponse 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	data	data	TaskInstanceStatusListData		0	

序号	名称	标识符	类型	长度	必选/ 可选	描述
5	code	code	String		0	状态码
6	message	message	String		0	状态消息
7	requestId	requestId	String		0	请求 ID

d) 请求示例

```
{
  "ids": [
    1
  ]
}
```

e) 响应示例

```
{
  "data": [
    {
      "id": 1,
      "status": "RUNNING",
      "onOff": "ON"
    }
  ],
  "code": "0",
  "message": "success",
  "requestId": "xxxx-xxxx-xxxx"
}
```

**附录 D**  
**(规范性)**  
**鉴权要求**

**D.1 基本要求**

本章节共规定了9个鉴权相关接口，主要包括鉴权的注册、删除和查询等，见表D.1。其中，算法算力平台与用户之间的鉴权接口调度流程见图D.1，各个接口属性信息应符合表D.2-C.10的规定，具体应符合章节D.2-C.10的规定。

**表 D.1 鉴权要求接口**

序号	URI	方法	功能说明
1	/ACP/authServer	POST	注册鉴权服务
2	/ACP/authServer/{provider}	DELETE	删除鉴权服务
3	/ACP/authServer/{provider}	PUT	修改鉴权服务
4	/ACP/authServer	GET	查询鉴权服务列表
5	/ACP/authServer/{provider}	GET	查询鉴权服务详情
6	/ACP/license	POST	导入授权文件
7	/ACP/license/{id}	DELETE	删除授权文件
8	/ACP/license	GET	查询授权文件列表
9	/ACP/license/{id}	GET	查询授权文件详情

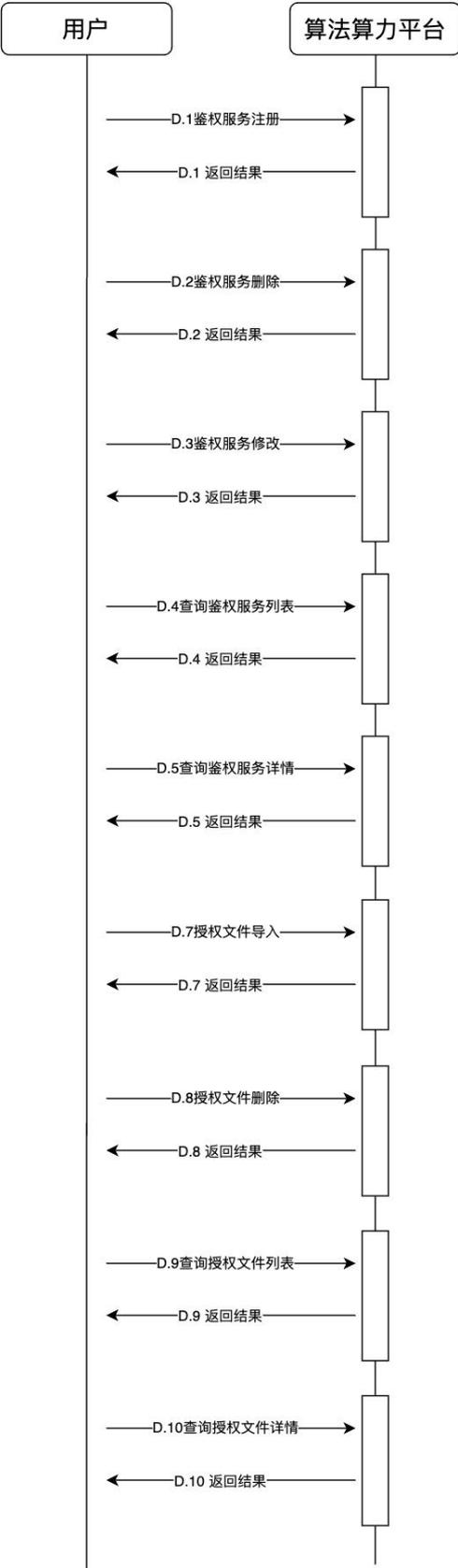


图 D.1 鉴权接口调用流程

表 D.2 注册鉴权服务

URI	/ACP/authServer		
功能	注册鉴权服务		
方法	查询字符串	请求消息体	响应消息体
POST	无	AuthServerRegisterRequest	AuthServerRegisterResponse
注释	注册鉴权服务到算法算力服务管理平台		

表 D.3 删除鉴权服务

URI	/ACP/authServer/{provider}		
功能	删除鉴权服务		
方法	查询字符串	请求消息体	响应消息体
DELETE	无		EmptyResponse
注释	从算法算力服务管理平台删除鉴权服务		

表 D.4 修改鉴权服务

URI	/ACP/authServer/{provider}		
功能	修改鉴权服务		
方法	查询字符串	请求消息体	响应消息体
PUT	无	AuthServerUpdateRequest	EmptyResponse
注释	鉴权服务修改		

表 D.5 查询鉴权服务列表

URI	/ACP/authServer		
功能	查询鉴权服务列表		
方法	查询字符串	请求消息体	响应消息体
GET	AuthServerListRequest		AuthServerListResponse
注释	查询鉴权服务列表		

表 D.6 查询鉴权服务详情

URI	/ACP/license/{id}		
功能	查询鉴权服务详情		
方法	查询字符串	请求消息体	响应消息体
GET			AuthServerDetailResponse
注释	查询鉴权服务详情		

表 D.7 导入授权文件

URI	/ACP/license		
功能	导入授权文件		
方法	查询字符串	请求消息体	响应消息体
POST		LicenseImportRequest	LicenseImportResponse
注释	导入授权文件到算法算力服务管理平台		

表 D.8 删除授权文件

URI	/ACP/license
-----	--------------

功能	删除授权文件		
方法	查询字符串	请求消息体	响应消息体
DELETE		LicenseDeleteRequest	EmptyResponse
注释	从算法算力服务管理平台删除授权文件		

表 D.9 查询授权文件列表

URI	/ACP/license		
功能	查询授权文件列表		
方法	查询字符串	请求消息体	响应消息体
GET	LicenseListRequest		LicenseListResponse
注释	查询授权文件列表		

表 D.10 查询授权文件详情

URI	/ACP/license/{id}		
功能	查询授权文件详情		
方法	查询字符串	请求消息体	响应消息体
GET			LicenseListResponse
注释	查询授权文件详情		

## D.2 注册鉴权服务

- a) URI: POST /ACP/authServer  
b) 请求参数:

表 D.11 AuthServerRegisterRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	鉴权服务器	server	String		R	鉴权服务器 ip:port, 可英文逗号分割多个
2	算法提供方	provider	String		0	算法提供方, 一个 provider 只允许注册一个 server

- c) 响应参数

表 D.12 AuthServerRegisterResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	ID		R	鉴权服务 ID
4	请求 id	requestId	String		0	请求 ID

表 D.13 ID 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	编号	id	Integer		R	授权记录 ID

d) 请求示例:

```
{
  "server": "192.108.1.1:8808,192.108.1.2:8808",
  "provider": "baidu"
}
```

e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": {
    "id": 1
  }
}
```

### D.3 删除鉴权服务

a) URI: DELETE /ACP/authServer/{provider}

b) 请求参数:

表 D.14 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	提供方	provider	String		R	提供方

c) 响应参数:

EmptyResponse

d) 响应示例:

```
{
  "code": "0",
  "message": "success"
}
```

### D.4 修改鉴权服务

a) URI: PUT /ACP/authServer/{provider}

b) 请求参数

表 D.15 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	提供方	provider	String		R	提供方

表 D.16 AuthServerUpdateRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	鉴权服务器	server	String		R	鉴权服务器 ip:port, 可英文逗号分割多个

c) 响应参数:

EmptyResponse

d) 请求示例:

```
{
  "server": "192.108.1.1:8808,192.108.1.2:8808"
}
```

e) 响应示例:

```
{
  "code": "0",
  "message": "success"
}
```

#### D.5 查询鉴权服务列表

a) URI: GET /ACP/authServer

b) 请求参数:

表 D.17 AuthServerListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法提供方	provider	String		0	算法提供方
2	创建时间查询开始点	createTimeFrom	String		0	创建时间查询开始点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22
3	创建时间查询结束点	createTimeTo	String		0	创建时间查询结束点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22

c) 响应参数

表 D.18 AuthServerListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	AuthServer[]		R	数据
4	请求 id	requestId	String		0	请求 ID

表 D.19 AuthServer 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	编号	id	Integer		R	授权记录 ID
2	鉴权服务器	server	String		R	鉴权服务器 ip:port, 可英文逗号分割多个
3	算法提供方	provider	String		0	算法提供方
4	创建时间	createDate	String		0	创建时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22

d) 请求示例:

```
{
  "provider": "baidu",
  "createTimeFrom": "2023-04-24 23:15:22",
  "createTimeTo": "2023-04-25 23:15:22"
}
```

e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": [
    {
      "id": 1,
      "server": "192.108.1.1:8808,192.108.1.2:8808",
      "provider": "baidu",
      "createdDate": "2023-04-24 23:15:22"
    }
  ]
}
```

#### D.6 查询鉴权服务详情

a) URI: GET /ACP/authServer/{provider}

b) 请求参数:

表 D.20 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	提供方	provider	String		R	提供方

c) 响应参数:

表 D.21 AuthServerDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	数据	data	AuthServer		R	数据
4	请求 id	requestId	String		0	请求 ID

d) 响应示例：

```
{
  "code": "0",
  "message": "success",
  "data": {
    "id": 1,
    "server": "192.108.1.1:8808,192.108.1.2:8808",
    "provider": "baidu",
    "createdDate": "2023-04-24 23:15:22"
  }
}
```

#### D.7 导入授权文件

- a) URI: POST /ACP/license  
b) 请求参数：

表 D.22 LicenseImportRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	授权文件	file	File		R	授权文件
2	算法提供方	provider	String		0	算法提供方

c) 响应参数

表 D.23 LicenseImportResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	数据	data	ID		R	授权记录 ID
4	请求 id	requestId	String		R	请求 ID

d) 请求示例：

Content-Type: multipart/form-data;  
file: license  
provider: baidu

e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data"
```

f) 请求示例:

```
{
  "provider": "baidu"
}
```

g) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": 1
}
```

#### D.8 删除授权文件

a) URI: DELETE /ACP/license/{id}

b) 请求参数

表 D.24 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	编号	id	Integer		R	授权记录 ID

c) 响应参数

EmptyResponse

d) 响应示例

```
{
  "code": "0",
  "message": "success"
}
```

#### D.9 查询授权文件列表

a) URI: GET /license

b) 请求参数

表 D.25 LicenseListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法提供方	provider	String		0	算法提供方
2	创建时间查询开始点	createTimeFrom	String		0	创建时间查询开始点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123

序号	名称	标识符	类型	长度	必选/可选	说明
3	创建时间查询结束点	createTimeTo	String		0	创建时间查询结束点 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123

## c) 响应参数

表 D. 26 LicenseListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码(0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	数据	data	License []		R	数据
4	请求 id	requestId	String		0	请求 ID

表 D. 27 License 属性

序号	标识符	名称	必选/可选	类型	长度	说明
1	id	编号	R	Integer		授权记录 ID
2	provider	算法提供方	0	String		算法提供方
3	createdDate	创建时间	0	String		创建时间 yyyy-MM-dd HH:mm:ss.sss, 例: 2019-06-01 23:15:22.123

## d) 请求示例

```
{
  "provider": "baidu",
  "createTimeFrom": "2023-04-24 23:15:22.123",
  "createTimeTo": "2023-04-25 23:15:22.123"
}
```

## e) 响应示例

```
{
  "code": "0",
  "message": "success",
  "data": [
    {
      "id": 1,
      "provider": "baidu",
      "createdDate": "2023-04-24 23:15:22"
    }
  ]
}
```

```

    }
  ]
}

```

#### D.10 查询授权文件详情

- a) URI: GET /license/{id}
- b) 请求参数

表 D.28 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	编号	id	Integer		R	授权记录 ID

表 D.29 响应参数属性

序号	标识符	名称	必选/可选	类型	长度	说明
1	code	结果状态码	R	String		结果状态码 (0: 成功, 其他: 失败)
2	message	结果描述	R	String		结果描述
3	data	数据	R	License		数据
4	requestId	请求 id	R	String		请求 ID

- c) 响应示例:

```

{
  "code": "0",
  "message": "success",
  "data": {
    "id": 1,
    "provider": "baidu",
    "createdDate": "2023-04-24 23:15:22"
  }
}

```

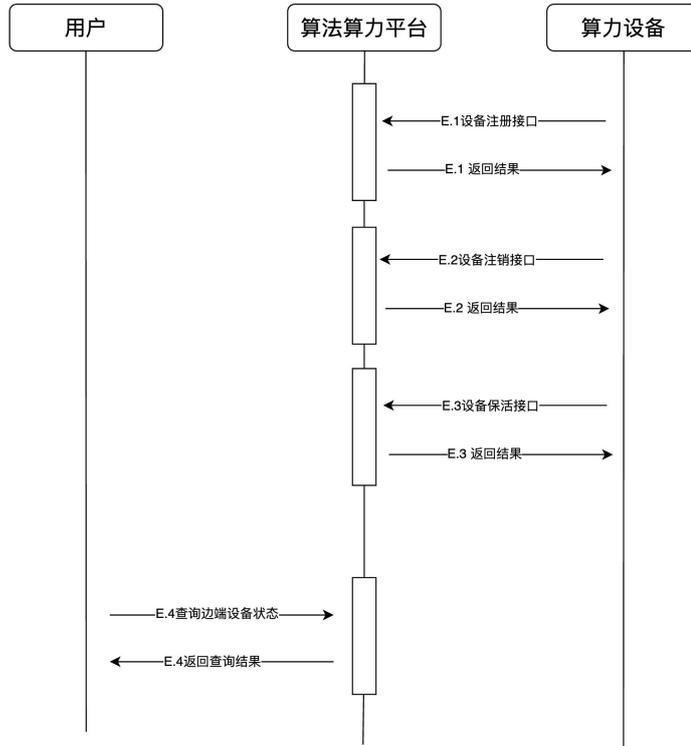
附录 E  
(规范性)  
设备算力接入

E.1 基本要求

本章节共规定了4个设备算力接入接口，主要包括算力设备的注册、注销、保活等，见表E.1。其中，算法算力平台与用户和算力设备之间的接口调度流程见图E.1，各个接口属性信息应符合表E.2-E.5的规定，具体应符合章节E.2-E.5的规定。

表 E.1 算力设备接入接口

序号	URI	方法	功能说明
1	/ACP/device/register	POST	注册算力设备
2	/ACP/device/unregister	POST	注销算力设备
3	/ACP/device/keepalive	POST	保活算力设备
4	/ACP/device/status/{deviceCode}	GET	查询边端设备的物理状态及资源占用情况



图E.1 设备算力接入接口流程

表 E.2 注册算力设备

URI	/ACP/device/register		
功能	注册算力设备		
方法	查询字符串	请求消息体	响应消息体
POST	无	RegisterRequest	EmptyResponse
注释	算力设备注册		

表 E.3 注销算力设备

URI	/ACP/device/unregister
-----	------------------------

功能	注销算力设备		
方法	查询字符串	请求消息体	响应消息体
POST	无	UnRegisterRequest	EmptyResponse
注释	算力设备注销		

表 E.4 保活算力设备

URI	/ACP/device/keepalive		
功能	保活算力设备		
方法	查询字符串	请求消息体	响应消息体
POST	无	KeepaliveRequest	EmptyResponse
注释	算力设备保活		

表 E.5 查询算力设备状态

URI	/ACP/device/status/{deviceCode}		
功能	查询算力设备状态		
方法	查询字符串	请求消息体	响应消息体
GET	无		DeviceStatusResponse
注释	查询算力设备状态		

## E.2 注册算力设备

- a) POST /ACP/device/register  
b) 请求参数

表 E.6 RegisterRequest 属性

序号	参数名称	必填	类型	说明
1	code	R	String	编码 code
2	ip	R	String	端设备算法 API 服务 ip
3	port	R	Integer	端设备算法 API 服务 port
4	username	0	String	API 服务用户名
5	password	0	String	宜采用国密 SM2 加密算法对密码加密后传输

- c) 响应参数  
EmptyResponse  
d) 请求体示例

```
{
  "deviceCode": "device_code",
  "type": "EDGE_COMPUTE_DEVICE",
  "ip": "127.0.0.1",
  "port": 8123,
  "username": "username",
  "password": "password"
}
```

- e) 响应体示例:

```
{
  "data": null,
  "code": "0",
}
```

```

    "message": "success"
}
    
```

E.3 注销算力设备

- a) POST /ACP/device/unregister
- b) 请求参数

表 E.7 UnRegisterRequest 属性

序号	参数名称	必填	类型	说明
1	code	R	String	编码 code
2	type	R	String	类型 EDGE_COMPUTE_DEVICE: 边缘设备 FRONT_END_DEVIE: 前端设备
3	username	R	String	用户名
4	password	R	String	密码

- c) 响应参数

EmptyResponse

- d) 请求体示例

```

{
  "deviceCode": "device_code",
  "type": "EDGE_COMPUTE_DEVICE",
  "username": "username",
  "password": "password"
}
    
```

- e) 响应体示例:

```

{
  "data": null
  "code": "0",
  "message": "success"
}
    
```

E.4 保活算力设备

边缘设备需实现设备保活接口，定期向平台发送保活信息，超时时间内没收到保活信息会认为设备丢失。

- a) POST /ACP/device/keepalive
- b) 请求参数

表 E.8 KeepaliveRequest 属性

序号	参数名称	必填	类型	说明
1	code	R	String	编码 code
2	expires	R	Integer	超时时间，单位秒

- c) 响应参数

EmptyResponse

- d) 请求体示例

```

{
    
```

```
"deviceCode": "device_code",
"expires":60
}
```

e) 响应体示例

```
{
"data": null
"code": "0",
"message": "success"
}
```

### E.5 查询设备状态及资源

- a) GET /ACP/device/status/{deviceCode}  
b) 响应参数

表 E.9 DeviceStatusResponse 属性

序号	参数名称	必填	类型	说明
1	code	R	String	结果状态码（0: 成功, 其他: 失败）
2	message	R	String	请求结果描述
3	data	0	DeviceInfo	边端设备状态信息

表 E.10 DeviceInfo 属性

序号	参数名称	必填	类型	说明
1	id	R	Long	边端设备 id
2	deviceCode	R	String	边端设备 code
3	isOnline	R	Boolean	是 0 在线
4	cpuUtilization	0	String	cpu 利用率, 0.11 表示 11%
5	memUsage	0	String	内存占用, 单位 GB
6	memFree	0	String	内存剩余, 单位 GB
7	diskUsage	0	String	磁盘占用, 单位 GB
8	diskFree	0	String	磁盘剩余, 单位 GB

c) 请求体示例

```
{
"deviceCode": "device_code"
}
```

d) 响应体示例:

```
{
  "data": {
    "id": 1
    "deviceCode": "device_code",
    "isOnline": true,
    "cpuUtilization": "0.11",
    "memUsage": "24.395",
    "memFree": " 39.605 "
    "diskUsage": "128.1",
    "diskFree": "127.9"
  },
  "code": "0",
  "message": "success"
}
```

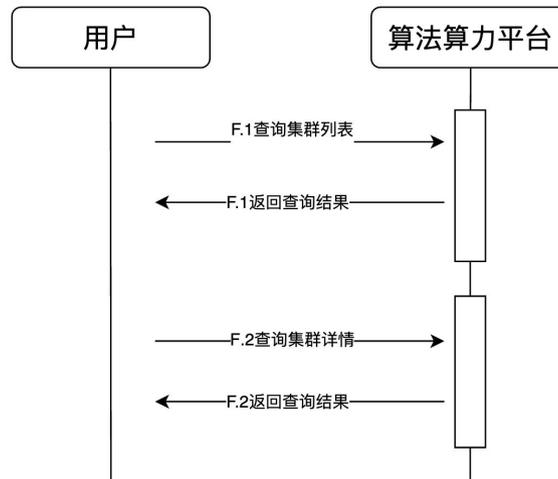
附录 F  
(规范性)  
中心算力查询接口

### F.1 基本要求

本章节共规定了2个中心算力查询接口，主要包括中心算力查询接口，见表F.1。其中，算法算力平台与用户之间的集群查询接口调度流程见图F.1，各个接口属性信息应符合表F.2-F.3的规定，具体应符合章节F.2-F.3的规定。

表 F.1 中心算力查询接口

序号	URI	方法	功能说明
1	/ACP/cluster	GET	查询集群列表
2	/ACP/cluster/{clusterCode}	GET	查询集群详细信息



图F.1 中心算力查询接口调用流程

表 F.2 查询集群列表

URI	/ACP/cluster		
功能	查询集群列表		
方法	查询字符串	请求消息体	响应消息体
GET	无	ClusterListRequest	ClusterListResponse
注释	在算法算力服务管理平台查询集群列表		

表 F.3 查询集群详情

URI	/ACP/cluster/{clusterCode}
功能	查询集群详情

方法	查询字符串	请求消息体	响应消息体
GET	id	无	ClusterDetailResponse
注释	在算法算力服务管理平台查询集群详情		

## F.2 查询集群列表

- a) URI: POST /ACP/cluster  
 b) 请求参数:

表 F.4 ClusterListRequest 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	集群名称	clusterName	String		0	集群名称
2	集群操作状态	status	String		0	集群操作状态 (Creating, CreateSuccess, CreateFail, Deleting, DeleteSuccess, DeleteFail)
3	分页参数	pageRequest	ApiPageRequest		R	分页参数 pageNumber=1, pageSize=10

- c) 响应参数:

表 F.5 ClusterListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		<u>R</u>	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		<u>R</u>	结果描述
3	数据	data	ClusterPageResponse		<u>R</u>	分页结果数据
4	请求 id	requestId	String		<u>0</u>	请求 ID

表 F.6 ClusterPageResponse 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	集群列表	pageContent	ClusterInfo[]		R	
2	分页信息	pageRequest	ApiPageRequest		R	
3	集群总数	total	Integer		R	
4	总页数	totalPages	Integer		R	

表 F.7 ClusterInfo 属性

序号	名称	标识符	类型	长度	必选/ 可选	描述
1	集群代码	clusterCode	String		R	集群代码
2	集群名称	clusterName	String		R	集群名称
3	k8s 版本	k8sVersion	String		R	k8s 版本
4	集群操作状态	status	String		R	集群操作状态 (Creating, CreateSuccess, CreateFail, Deleting, DeleteSuccess, DeleteFail)
5	集群运行状态	clusterHealthStatus	String		0	集群运行状态, Healthy=运行中; Unhealthy=不健康; RunningFault=集群故障

## d) 请求示例

```
{
  "clusterName": "",
  "clusterStatus": "",
  "pageRequest": {
    "pageNumber": 1,
    "pageSize": 20,
    "orderList": [
      {
        "property": "id",
        "direction": "desc"
      }
    ]
  }
}
```

## e) 响应示例

```
{
  "data": {
    "pageContent": [
      {
        "clusterName": "",
        "k8sVersion": "",
        "clusterCode": "",
        "status": "",
        "clusterHealthStatus": ""
      }
    ],
    "pageRequest": {
      "pageNumber": 1,
      "pageSize": 20,
      "orderList": [
        {
          "property": "id",
```

```

        "direction": "desc"
    }
  ]
},
"total": 1
},
"code": "0",
"message": "success",
"requestId": "xxxx-xxxx-xxxx"
}

```

F.3 查询集群详情

- a) URI: POST /ACP/cluster/{clusterCode}
- b) 请求参数

表 F.8 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	集群代码	clusterCode	String		R	集群代码

- c) 响应参数:

表 F.9 ClusterDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	状态码	code	String		R	状态码
2	状态消息	message	String		R	状态消息
3	集群详情	data	CluterInfo		0	
4	请求 ID	requestId	String		0	请求 ID

表 F.10 ClusterInfo 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	集群代码	clusterCode	String		R	集群代码
2	集群名称	clusterName	String		0	集群名称
3	k8s 版本	k8sVersion	String		0	k8s 版本
4	集群操作状态	status	String		0	集群操作状态 (Creating, CreateSuccess, CreateFail, Deleting, DeleteSuccess, DeleteFail)
5	集群运行状态	clusterHealthStatus	String		0	集群运行状态, Healthy=运行中; Unhealthy=不健康; RunningFault=集群故障

d) 请求示例

```
{  
  "clusterCode": ""  
}
```

e) 响应示例

```
{  
  "data": {  
    "clusterName": "",  
    "k8sVersion": "",  
    "clusterCode": "",  
    "status": "",  
    "statusInfo": "",    "k8sToken": "",  
    "clusterHealthStatus": "",  
    "clusterIngressAddress": "",  
  },  
  "code": "0",  
  "message": "success",  
  "requestId": "xxxx-xxxx-xxxx"  
}
```

## F.3 查询机器列表

- a) URI: POST /ACP/machine/{clusterCode}  
 b) 请求参数

表 F.11 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	集群代码	clusterCode	String		R	集群代码

- c) 响应参数:

表 F.12 MachineListResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	状态码	code	String		R	状态码
2	状态消息	message	String		R	状态消息
3	集群详情	data	MachineInfo[]		0	
4	请求 ID	requestId	String		0	请求 ID

表 F.13 MachineInfo 属性

序号	名称	标识符	类型	长度	必选/可选	描述
1	机器代码	machineCode	String		0	机器代码
2	机器名称	machineName	String		0	机器名称
3	机器 ip	machineIp	String		0	机器 ip
4	机器虚拟 ip	machineVip	String		0	机器虚拟 ip, 外部服务访问本机的 ip
5	机器配置信息	machineInfo	String		0	机器配置信息
6	cpu 类型	cpuType	String		0	cpu 类型
7	cpu	cpuAmount	Integer		0	cpu 信息 核数
8	加速卡类型	xpuType	String		0	加速卡类型, 参考附录 I
9	加速卡数量	xpuAmount	Integer		0	加速卡卡数量
10	加速卡内存	xpuMemory	Integer		0	加速卡内存
11	加速卡内存单位	xpuMemoryUnit	String		0	加速卡内存单位, 默认为 G
12	内存	memoryAmount	Integer		0	内存
13	硬盘	diskAmount	String		0	硬盘
14	机器所属的集群	clusterCode	String		0	机器所属的集群, 空为不在集

						群中
15	机器在集群中的角色	clusterRole	String		0	机器在集群中的角色 (MASTER, SLAVE)
16	机器在集群中的操作状态	clusterNodeStatus	String		0	机器在集群中的操作状态 (, ADDING, ADD_SUCCESS, ADD_FAILURE, DELETING, DELETE_SUCCESS, DELETE_FAILURE)
17	机器在集群中的运行状态 (Ready, NotReady, NotFound)	nodeRunningStatus	String		0	机器在集群中的运行状态 (Ready, NotReady, NotFound)
18	机器在线离线状态 (OFFLINE, ONLINE)	status	String		0	机器在线离线状态 (OFFLINE, ONLINE)

d) 请求示例

```
{
  "clusterCode": ""
}
```

e) 响应示例

```
{
  "data": [
    {
      "machineCode": "",
      "machineName": "",
      "machineIp": "",
      "machineVip": "",
      "machineInfo": "",
      "cpuType": "",
      "cpuAmount": 1,
      "xpuType": "",
      "xpuAmount": 1,
      "xpuMemory": 1,
      "xpuMemoryUnit": "",
      "memoryAmount": 1,
      "diskAmount": "",
      "clusterCode": "",
      "clusterRole": "",
      "clusterNodeStatus": "",
      "nodeRunningStatus": "",
      "status": ""
    }
  ],
  "code": "0",
  "message": "success",
}
```

T/BAX XXXX.5—2023

```
"requestId": "xxxx-xxxx-xxxx"  
}
```

附录 G  
(规范性)  
边端设备鉴权接口

### G.1 基本要求

本章节共规定了2个边端设备鉴权接口，主要包括边端设备查询和授权查询接口，见表G.1。其中，算法算力平台与边端设备和用户之间的鉴权接口调度流程见图G.1，各个接口属性信息应符合表G.2-G.3的规定，具体应符合章节G.2-G.3的规定。

表 G.1 边端设备鉴权接口

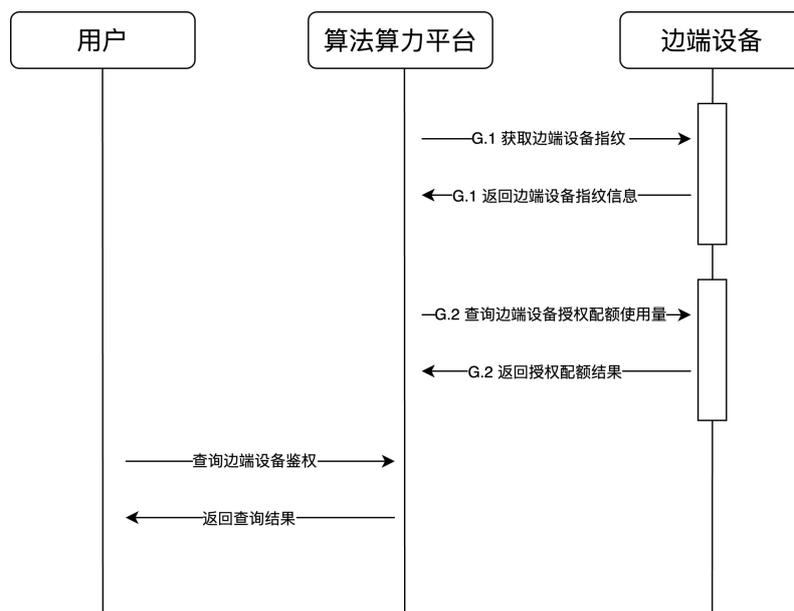
序号	URI	方法	功能说明
1	/ACP/device/{deviceCode}/fingerPrint	GET	平台查询边端设备的指纹
2	/ACP/device/{deviceCode}/{algoPackageCode}/quota	GET	边端设备提供授权配额量查询

表 G.2 获取边端设备指纹

URI	/ACP/device/{deviceCode}/fingerPrint		
功能	获取边端设备指纹		
方法	查询字符串	请求消息体	响应消息体
GET	无	无	DeviceFingerPrintResponse
注释	边端设备需实现指纹获取接口，供平台查询设备指纹		

表 G.3 查询边端设备授权配额使用量

URI	/ACP/device/{deviceCode}/{algoPackageCode}/quota		
功能	查询边端设备授权配额使用量		
方法	查询字符串	请求消息体	响应消息体
GET	无	无	DeviceQuotaDetailResponse
注释	边端设备需要提供授权配额量查询接口，供平台查询指定算法包的已使用配额和总配额		



图G.1 边端设备鉴权接口调用流程

## G.2 获取边端设备指纹的接口

- a) URI: GET /ACP/device/{deviceCode}/fingerPrint  
 b) 请求参数

表 G.4 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	边端设备编码	deviceCode	String		R	边端设备编码

- c) 响应参数:

表 G.5 响应参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	指纹数据	data	String		0	设备指纹
4	请求 id	requesttId	String		0	请求 ID

- d) 请求示例:

GET /ACP/device/device-1/fingerPrint

- e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": "xxx-xxx-xxx-xxx"
}
```

## G.3 查询边端设备授权配额使用量

- a) URI: GET /ACP/device/{deviceCode}/{algoPackageCode}/quota  
 b) 请求参数:

表 G.6 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法包编码	deviceCode	String		R	算法包编码
2	算法包编码	algoPackageCode	String		R	算法包编码

- c) 响应参数:

表 G.7 DeviceQuotaDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码（0：成功，其他：失败）
2	结果描述	message	String		R	结果描述
3	指纹数据	data	DeviceQuota		R	配额信息
4	请求 id	requestId	String		0	请求 ID

表 G.8 DeviceQuota 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法包编码	algoPackageCode	String		0	算法包编码
2	已使用配额	quotaUsed	Integer		R	已使用配额，路数
3	总配额	quotaTotal	Integer		R	总配额，路数

d) 请求示例：

GET /ACP/device/device-1/BAIDU-HUMAN\_ATTR\_ALERT-1.0.0/quota

e) 响应示例：

```
{
  "code": "0",
  "message": "success",
  "data": {
    "algoPackageCode": "human-struct",
    "quotaUsed": 3,
    "quotaTotal": 30
  }
}
```

附录 H  
(规范性)  
中心鉴权服务接口

H.1 基本要求

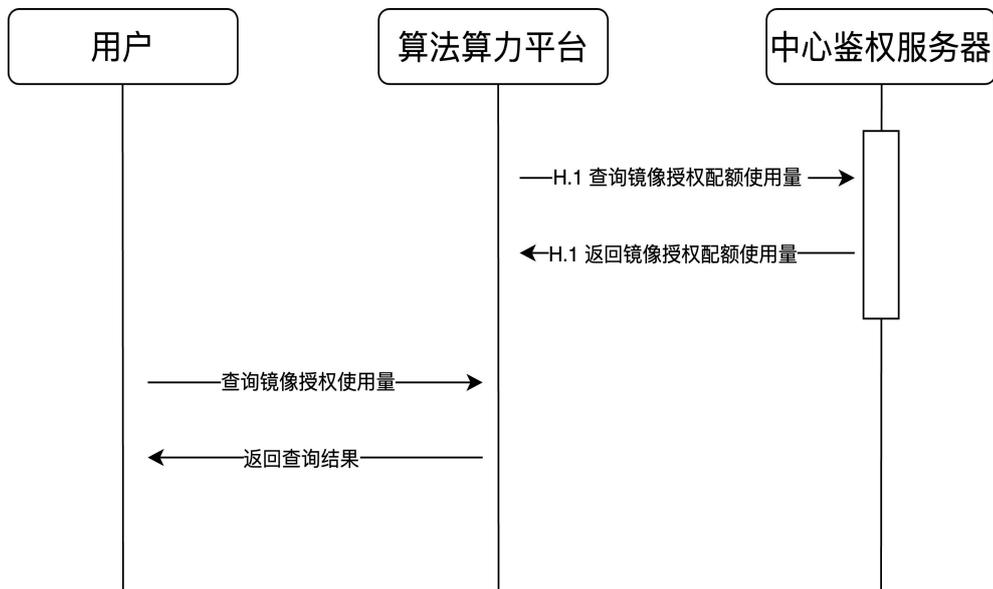
本章节共规定了1个中心鉴权服务查询接口，见表H.1。其中，算法算力平台与用户和中心鉴权服务器之间的鉴权接口调度流程见图H.1，接口属性信息应符合表H.2的规定，具体应符合章节H.2的规定。

表 H.1 中心鉴权服务接口

序号	URI	方法	功能说明
1	/ACP/auth/quota/query/{engineCode}	GET	供平台查询指定算法引擎的已使用配额和总配额。

表 H.2 查询算法引擎授权配额使用量

URI	/ACP/auth/quota/query		
功能	查询算法引擎授权配额使用量		
方法	查询字符串	请求消息体	响应消息体
GET	无	无	EngineQuotaDetailResponse
注释	中心鉴权服务器需要提供算法引擎的授权配额使用量查询接口，供平台查询指定算法引擎的已使用配额和总配额。		



图H.1 中心鉴权接口调用流程

H.2 查询算法引擎授权配额使用量

- a) URI: GET /ACP/auth/quota/query/{engineCode}
- b) 请求参数:

表 H.3 请求参数属性

序号	名称	标识符	类型	长度	必选/可选	说明
----	----	-----	----	----	-------	----

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法引擎编码	engineCode	Integer		R	算法引擎编码

c) 响应参数:

表 H.4 EngineQuotaDetailResponse 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	结果状态码	code	String		R	结果状态码 (0: 成功, 其他: 失败)
2	结果描述	message	String		R	结果描述
3	配额信息	data	EngineQuota		R	配额信息
4	请求 ID	requestId	String		0	请求 ID

表 H.5 EngineQuota 属性

序号	名称	标识符	类型	长度	必选/可选	说明
1	算法引擎编码	engineCode	String		R	算法引擎编码
2	已使用配额	quotaUsed	Integer		R	已使用配额, 路数
3	总配额	quotaTotal	Integer		R	总配额, 路数

d) 请求示例:

```
{
  "engineCode": "engine-1"
}
```

e) 响应示例:

```
{
  "code": "0",
  "message": "success",
  "data": {
    "engineCode": "engine-1",
    "quotaUsed": 3,
    "quotaTotal": 30
  }
}
```

附 录 I  
(规范性)  
加速卡类型

本章节主要规定了加速卡的类型和名称，详情见表I.1。

表 I.1 加速卡类型

序号	加速卡类型	加速卡名称
1	NVIDIA_T4	英伟达-t4
2	NVIDIA_P4	英伟达-p4
3	BITMAIN_1682	比特大陆 1682
4	BITMAIN_1684	比特大陆 1684
5	MLU370_S4	寒武纪 370S4
6	ASCEND_310	昇腾 310
7	KUNLUN_K200	昆仑 K200
8	CLOUDBLAZER_I10	燧原 i10
9	HAIGUANG_DCU1	海光 DCU 一号
10	KUNLUN_R200	昆仑 R200

参 考 文 献

---