**modserv-shim 技术设计文档**

1.1 **核心定位**

轻量级大模型引擎部署与全生命周期管控中间件

* 核心价值：以极简接入成本实现大模型引擎的 “一键部署、状态可视、动态运维”，支持多环境部署（K8s 原生支持，Docker 等环境插件化扩展）。
* 四层架构，形成 "请求接入→流程编排→资源操作→状态反馈" 闭环：

2. **架构详情**

2.1.1 **接入层**

* 基于 GIN 的 HTTP 服务，提供 RESTful 接口（部署 / 更新 / 删除 / 查询）
* 集成 Swagger 文档与 metrics 指标（部署成功率、资源使用率等）

2.1.2 **核心服务层**

* DeployManager：接收请求，调度对应流水线（部署 / 更新 / 删除），支持灰度、滚动更新策略
* StateManager：基于 FSM 管理引擎状态（Pending→Deploying→Running→Failed），异常状态转变记录
* TemplateManager：管理部署模板（K8s/Docker），支持参数动态注入
* ConfigManager：存储全局 / 实例配置，支持版本回滚

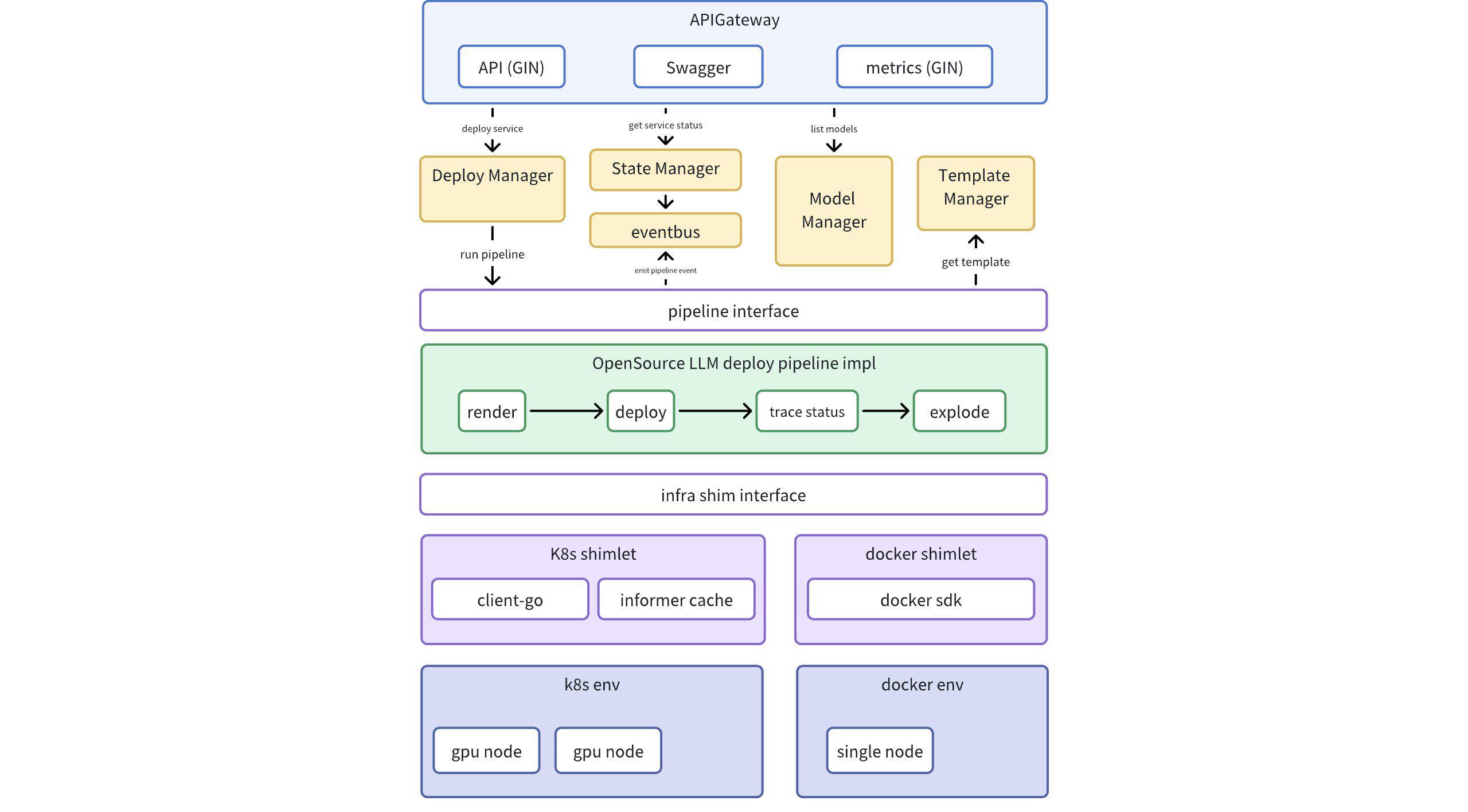
2.1.3 **流水线层**

* 抽象通用流程接口，按步骤编排操作（如部署流水线：校验→渲染模板→创建资源→健康检查）
* 支持步骤回滚与自定义钩子注入

2.1.4 **平台适配层**

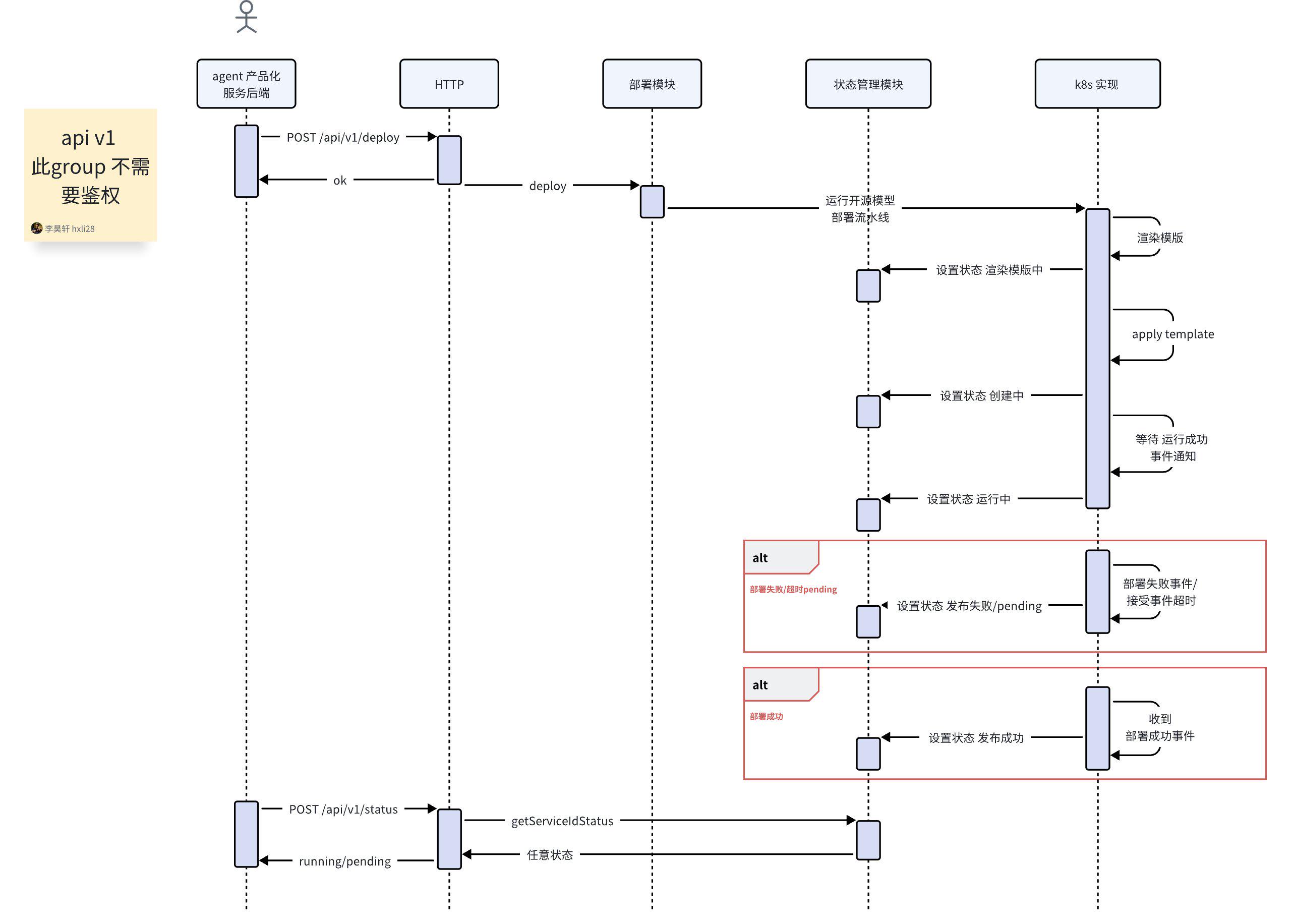
* 统一接口屏蔽环境差异，实现 K8s（client-go）、Docker（SDK）适配
* 插件化扩展新环境（如云 ECS）

2.1.1 **组件架构图**

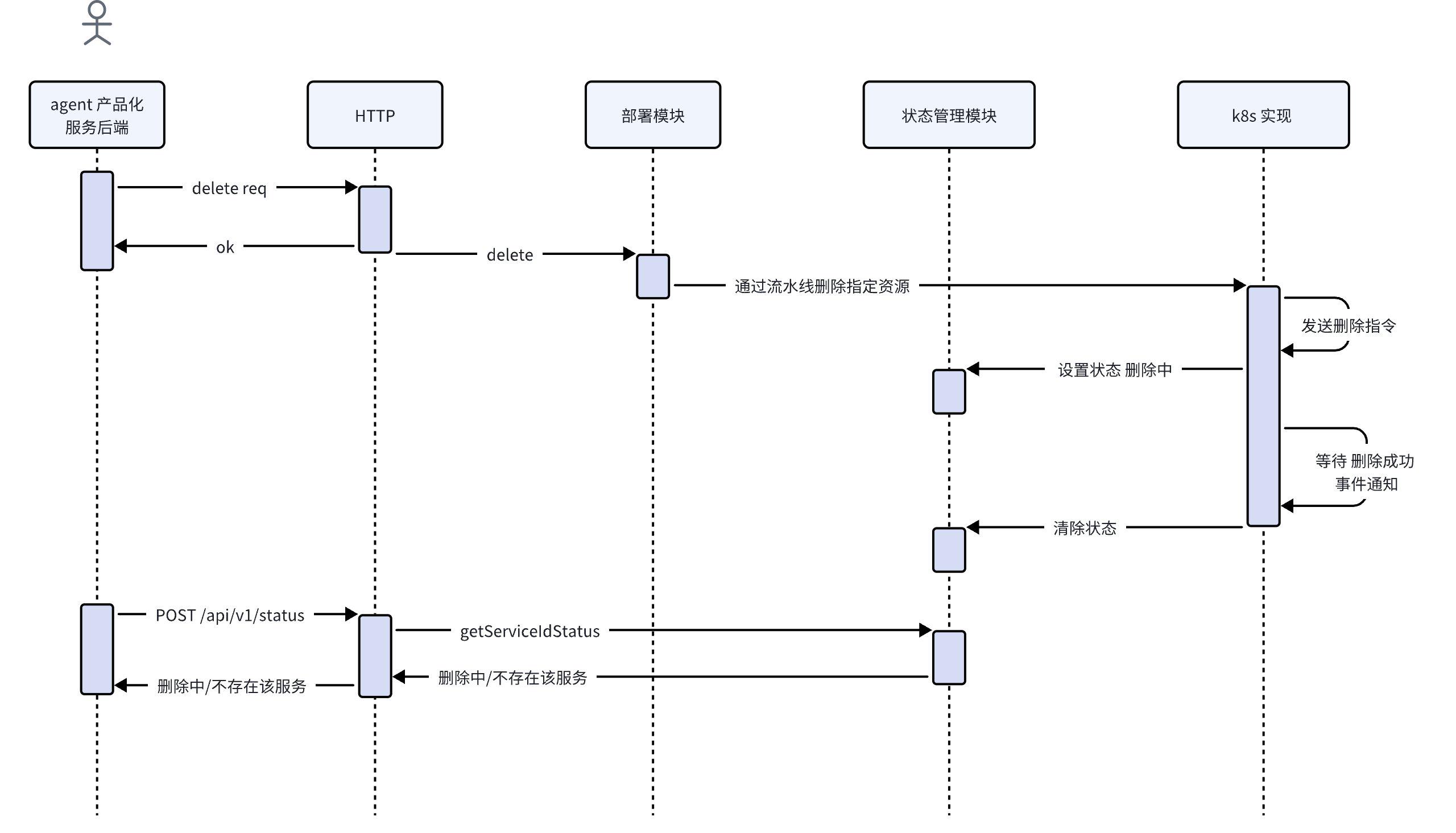


2.1.2 **启动时序图:**

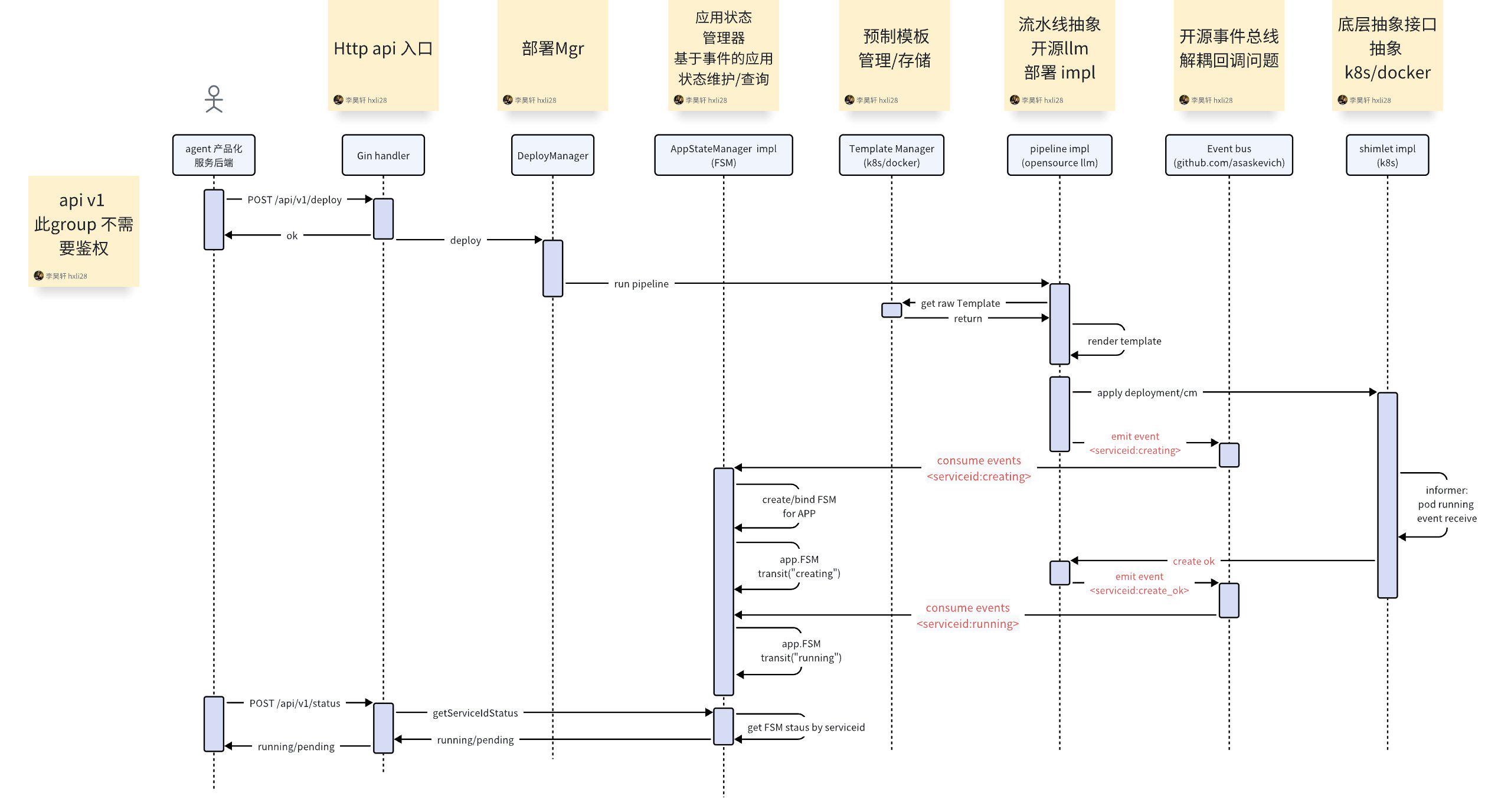
2.1.3 **业务部署/更新时序图(简化版):**



2.1.4 **业务删除时序图(简化版):**



2.1.5 **技术实现细节时序图:**



鉴权、错误处理缺失

2.3 **接口列表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口路径 | 请求方法 | 功能描述 | 核心请求参数 | 响应结果 |
| /api/v1/modserv/deploy | POST | 创建应用部署 | 名称、镜像、端口、平台类型 | 应用 ID、当前状态、操作结果 |
| /api/v1/modserv/{serviceid} | GET | 查询应用状态 | 应用名称 | 应用全量信息、当前状态 |
| /api/v1/modserv/{serviceid} | PUT | 更新应用配置 | 镜像、副本数、环境变量 | 操作结果、更新后状态 |
| /api/v1/modserv/{serviceid} | DELETE | 销毁应用 | 应用名称 | 操作结果 |
| /api/v1/modserv/modelsfiles | GET | List 约定目录下的模型 | 无 | 指定目录下的全量模型列表 |

2.3.1 **发布服务（创建新服务, 颁发 serviceid）**

* 请求方法：POST
* 请求 URL：/api/v1/modserv/deploy
* 请求头：

|  |
| --- |
| JSON {"Content-Type": "application/json"} |

* 请求体：

|  |
| --- |
| JSON {  "modelFile": "path/to/new/model/file",  "resourceRequirements": {  "acceleratorType": "NVIDIA H20", // 显卡类型更新为H20  "acceleratorCount": 2  },  "replicaCount": 1,  "env": [{"key": "BATCH\_SIZE", "value": "16"}] } |

* 响应体

|  |
| --- |
| JSON {  "code": 0,  "message": "服务创建成功",  "data": {"serviceId": "xdeepseekv3-<UUID>"} } |

2.3.2 **更新服务**

* 请求方法：PUT
* 请求 URL：/api/v1/modserv/xdeepseekv3-<UUID>（通过 URL 定位资源）
* 请求头：

|  |
| --- |
| JSON {"Content-Type": "application/json"} |

* 请求体：

|  |
| --- |
| JSON {  "modelFile": "path/to/updated/model/file",  "resourceRequirements": {  "acceleratorType": "NVIDIA H20", // 显卡类型更新为H20  "acceleratorCount": 2  },  "replicaCount": 2,  "env": [{"key": "BATCH\_SIZE", "value": "32"}] } |

* 响应体

|  |
| --- |
| JSON {  "code": 0,  "message": "服务更新成功",  "data": {"serviceId": "service-20231001-001", "status": "updating"} } |

2.3.3 **删除服务**

* 请求方法：DELETE（语义：删除指定资源）
* 请求 URL：/api/v1/modserv/xdeepseekv3-<UUID>（通过 URL 定位资源）
* 请求体：无需任何内容（资源标识已通过 URL 传递）

2.3.4 **可部署服务列表**

* 请求方法：GET
* 请求 URL：/api/v1/modserv/modellist
* 响应体： (返回模型名称 路径 的模型列表)

|  |
| --- |
| JSON {  "code": 0,  "message": "success",  "data": [  {  "modelName": "Llama-3-8B",  "modelPath": "/models/llama-3-8b",  },  {  "modelName": "BGE-Large",  "modelPath": "/models/bge-large",  }  ] } |