WiFi 音箱 - 量产版 AI 接入与商业化方案设计 v1

版本：1.0

日期：2026 年 3 月 21 日

目录

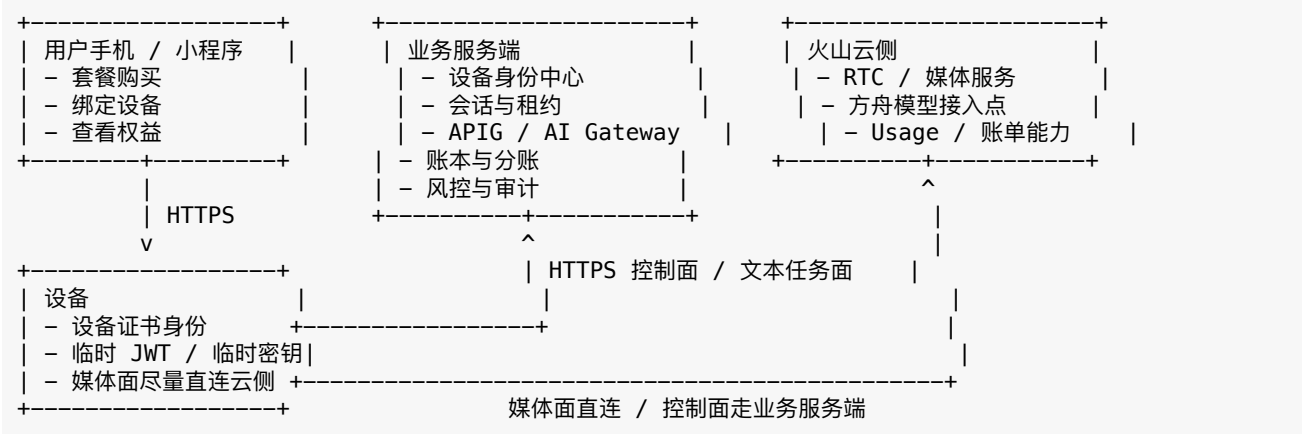
一、通用设计规范

1.1 设计目标

本方案面向量产 WiFi 音箱类设备，目标是在不明显牺牲用户体感的前提下，统一解决设备身份、AI 任务接入、套餐控制、权威计量、风险控制和商业化扩展问题。

目标	说明
安全可控	设备不长期持有可直接消费高价值 AI 资源的主密钥或主能力。
用户体验稳定	支持低水位续权、自然边界收口、弱网恢复和 graceful stop。
商业化量产	支持套餐 entitlement、家庭设备、渠道分账和企业客户扩展。
运维可审计	计量、账本、回调、审计链路可追踪、可纠偏、可回放。

1.2 边界划分



说明：控制面、商业面、计量面统一由业务服务端承接；媒体面尽量直连火山云侧；文本与高价值 AI 任务优先经你的 APIG / AI Gateway。

1.3 架构基线

Gateway Base URL：

```
https://api.example-ai.com
```

统一前缀：

```
/api/v1
```

项	约定
编码	请求与响应统一使用 application/json; charset=utf-8
时间单位	所有时间戳统一使用 Unix 秒
身份模型	设备以 device_id + 设备证书或设备密钥对作为根身份
会话模型	所有高价值任务均挂在 session_id 和 lease_id 上
计量模型	设备估算只用于续权；最终结算只认平台侧权威计量

1.4 设备身份与证书约束

量产设备建议在出厂时即写入唯一 device_id 和设备私钥。私钥优先放在安全芯片或系统安全区，服务端只保存对应公钥或证书映射。

字段	说明
product_key	产品线标识，例如 WIFI_SPEAKER_PRO
device_id	设备唯一 ID，协议层和账本层的统一设备主键
device_public_key	设备公钥或证书公钥，用于登录校验
firmware_version	当前固件版本，用于灰度、风控和兼容性控制
risk_level	服务端风险等级标签，用于强制降级或切代理通道

1.5 临时凭证与网关约束

本方案优先使用“设备登录控制服务 -> 获取短期 JWT -> 通过 APIG / AI Gateway 调用”这一条主路径。临时 API Key 仅作为受控增强能力，不作为所有量产设备的默认主模式。

凭证类型	推荐角色	约束
短期 JWT	主路径访问凭证	只允许调用你的 APIG / AI Gateway；TTL 建议 10-30 分钟。
临时 API Key	少量受控增强场景	必须绑定指定 endpoint 或 bot；TTL 尽量短；不可长期常驻设备。
RTC Token	媒体面准入凭证	仅解决入房和媒体权限，不承担商业配额控制职责。

1.6 通用请求头

Header	必填	说明
Content-Type	✅	固定 application/json
Authorization	✅	Bearer <JWT>，设备登录后获取的短期访问凭证
X-Api-Version	✅	协议版本，固定 1
X-Product-Key	✅	产品标识，如 WIFI_SPEAKER_PRO
X-Device-Id	✅	设备唯一标识
X-Request-Nonce	✅	请求随机串，用于防重放

1.7 通用响应结构

成功响应：

```
{
  "code": 0,
  "message": "success",
  "request_id": "1ec0bb1d-5e06-4c7d-80d8-6bd9d47f4c6a",
  "server_time": 1774051200,
  "data": {}
}
```

失败响应：

```
{
  "code": 1402,
  "message": "lease quota exhausted",
  "request_id": "1ec0bb1d-5e06-4c7d-80d8-6bd9d47f4c6a",
  "server_time": 1774051200,
  "data": {
    "grace_allowed": true,
    "suggested_action": "FINISH_CURRENT_SEGMENT"
  }
}
```

1.8 错误码设计

code	HTTP	含义
0	200	成功
1000	400	参数错误
1200	404	设备未激活或未注册

1201	403	设备已禁用
1300	401	JWT 无效或已过期
1400	403	会话开启被拒绝
1401	409	会话冲突或单设备单活冲突
1402	403	租约额度不足
1500	429	请求过于频繁
1600	409	会话恢复失败
5000	500	服务端内部错误

1.9 权威计量与账本对账

为补强会话审计、商业结算和异常排查，服务端应结合火山侧 usage、账单与数据回流能力，以平台侧权威计量作为最终结算依据。

来源	用途
设备 heartbeat / report	运行态感知、弱网分析、体验追踪
网关请求日志	链路追踪、风控审计、接口 SLA
平台 usage / 账单	最终扣费、对账和纠偏
数据回流	二级审计、质检、训练和事件复盘

1.10 套餐 entitlement 语义

面向用户售卖的是 entitlement 和能力包，而不是原始 token。内部统一换算为 budget_units，并在账本中记录预占、消耗、释放和修正。

外部售卖项	内部映射
月度陪伴时长	语音任务预算 units
月度故事次数	长文本片段预算 units
高级模型权限	model_tier entitlement
家庭共享设备数	max_bound_devices entitlement

1.11 低水位续权与 grace quota

策略	推荐值
文本任务低水位阈值	剩余 20%-30% 时后台续权

语音任务低水位阈值	剩余 30%-40% 时后台续权
grace quota	仅用于收完当前句 / 当前段 / 当前故事片
续权形态	静默后台续租，不阻塞当前生成链路

所有长故事、长播报、长续写任务都必须按自然边界切分，避免“生成到一半突然断”带来的糟糕体验。

1.12 风控与审计基线

规则	说明
单设备单活会话	同一设备在同一时刻只允许一个主会话
单用户并发设备数限制	防止家庭号或共享号绕过套餐设计
租约续签频率限制	防止异常续权放大账单风险
黑白名单和风险标签	高风险设备可强制切换为代理通道或直接拒绝
审计链路保留	会话、租约、回调、账本流水必须可追溯

二、系统总览

2.1 分层架构

2.2 关键对象

对象	说明
device_id	设备唯一身份主键
session_id	一段连续的用户使用会话
lease_id	一次任务租约或预算切片对象
request_id	链路追踪请求 ID
task_id	上游任务主键，用于 usage 与回流对账
budget_units	内部统一预算单位，用于 entitlement 和账本

2.3 会话状态机

```
IDLE
  |
  +--> OPENING --> ACTIVE --> DRAINING --> CLOSED
                    |   ^         |
                    |   |         +--> CLOSED_BY_QUOTA
                    |   |
                    |   +--> RENEWING
                    |
                    +--> INTERRUPTED --> RESUMING --> ACTIVE
                    |
                    +--> TERMINATED_BY_RISK
```

说明：DRAINING 表示已不再发放新租约，但允许设备使用 grace quota 将当前片段收口；INTERRUPTED / RESUMING 用于弱网和进程重启场景。

2.4 套餐与分账策略

维度	建议字段
用户	user_id、plan_id、entitlement_set
设备	device_id、product_key、risk_level
渠道 / OEM	channel_id、tenant_id
成本维度	model_tier、feature_id、task_type
对账维度	task_id、session_id、lease_id、request_id

三、接口详细设计

3.1 设备激活

请求路径：

```
POST /api/v1/device/activate
```

请求参数：

字段	类型	必填	说明
timestamp	int	✅	请求发送时间
product_key	string	✅	产品标识
device_id	string	✅	设备唯一 ID

device_nonce	string	✅	随机串，用于防重放
device_signature	string	✅	设备签名
firmware_version	string	✅	当前固件版本

请求示例：

```
{
  "timestamp": 1774051200,
  "product_key": "WIFI_SPEAKER_PRO",
  "device_id": "dev_20260321_000001",
  "device_nonce": "9d2a4ce8",
  "device_signature": "<SIGNATURE>",
  "firmware_version": "1.0.0"
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_activate_0001",
  "server_time": 1774051200,
  "data": {
    "activation_status": "ACTIVATED",
    "device_profile_version": "2026032101"
  }
}
```

说明：

设备激活成功后，后续所有登录、会话和账本行为都绑定到同一 device_id。

如果设备已激活，服务端可返回当前有效 profile 版本，避免重复激活。

3.2 设备登录

请求路径：

```
POST /api/v1/device/login
```

请求参数：

字段	类型	必填	说明
timestamp	int	✅	请求发送时间
device_nonce	string	✅	随机串

device_signature	string	✅	设备签名
firmware_version	string	✅	当前固件版本

请求示例：

```
{
  "timestamp": 1774051200,
  "device_nonce": "64b3ad87",
  "device_signature": "<SIGNATURE>",
  "firmware_version": "1.0.0"
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_login_0001",
  "server_time": 1774051200,
  "data": {
    "access_token": "<JWT>",
    "expires_at": 1774053000,
    "refresh_before_seconds": 120
  }
}
```

说明：

主路径推荐返回短期 JWT，让设备统一经 APIG / AI Gateway 发起后续业务请求。

临时 API Key 仅在受控增强场景下由服务端另行下发，不应作为默认模式。

3.3 开启会话

请求路径：

```
POST /api/v1/session/open
```

请求参数：

字段	类型	必填	说明
timestamp	int	✅	请求发送时间
user_id	string	✅	当前用户 ID
task_type	string	✅	任务类型，如 STORY / TALK / CHAT

device_state	string	✅	设备当前状态
audio_codec	string	否	语音场景下的音频编码

请求示例：

```
{
  "timestamp": 1774051200,
  "user_id": "user_10001",
  "task_type": "STORY",
  "device_state": "IDLE",
  "audio_codec": "G711A"
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_open_0001",
  "server_time": 1774051200,
  "data": {
    "session_id": "sess_20260321_abcd1234",
    "lease_id": "lease_20260321_0001",
    "granted_units": 12000,
    "soft_threshold_units": 3600,
    "grace_units": 1200,
    "gateway_route": "story-primary",
    "rtc_token": "<RTC_TOKEN>",
    "rtc_expire_at": 1774052100
  }
}
```

说明：

服务端在此处完成 entitlement 校验、预算预占、单设备单活会话校验和风控判断。

长故事任务建议在这里明确分段目标，以便后续自然边界续权。

3.4 续签租约

请求路径：

```
POST /api/v1/lease/renew
```

请求参数：

字段	类型	必填	说明

timestamp	int	✅	请求发送时间
session_id	string	✅	当前会话 ID
lease_id	string	✅	当前租约 ID
estimated_consumed_units	int	✅	设备侧估算已消费预算
current_segment	string	否	当前任务片段标识

请求示例：

```
{
  "timestamp": 1774051680,
  "session_id": "sess_20260321_abcd1234",
  "lease_id": "lease_20260321_0001",
  "estimated_consumed_units": 8600,
  "current_segment": "story_part_03"
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_renew_0001",
  "server_time": 1774051680,
  "data": {
    "next_lease_id": "lease_20260321_0002",
    "granted_units": 10000,
    "soft_threshold_units": 3000,
    "grace_units": 1200
  }
}
```

说明：

续权动作应由设备后台静默触发，不应阻塞当前流式输出链路。

若服务端判定不再续租，应允许设备进入 DRAINING，并使用 grace quota 收完当前片段。

3.5 心跳

请求路径：

```
POST /api/v1/device/heartbeat
```

请求参数：

字段	类型	必填	说明
timestamp	int	✅	请求发送时间
session_id	string	否	当前会话 ID
device_state	string	✅	设备当前运行状态
firmware_version	string	✅	固件版本
uptime_seconds	int	✅	运行时长

请求示例：

```
{
  "timestamp": 1774051800,
  "session_id": "sess_20260321_abcd1234",
  "device_state": "ACTIVE",
  "firmware_version": "1.0.0",
  "uptime_seconds": 7200
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_heartbeat_0001",
  "server_time": 1774051800,
  "data": {
    "heartbeat_interval_seconds": 60,
    "server_action": "NONE",
    "message": null
  }
}
```

说明：

心跳只用于设备状态感知、动作下发和弱网分析，不作为最终计量真相。

3.6 恢复会话

请求路径：

```
POST /api/v1/session/resume
```

请求参数：

字段	类型	必填	说明

timestamp	int	✅	请求发送时间
session_id	string	✅	待恢复会话 ID
last_lease_id	string	✅	最后一次已知租约 ID
resume_reason	string	✅	恢复原因，如 NETWORK_RECOVERED

请求示例：

```
{
  "timestamp": 1774051860,
  "session_id": "sess_20260321_abcd1234",
  "last_lease_id": "lease_20260321_0002",
  "resume_reason": "NETWORK_RECOVERED"
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_resume_0001",
  "server_time": 1774051860,
  "data": {
    "resume_status": "ALLOWED",
    "effective_lease_id": "lease_20260321_0002",
    "grace_allowed": true
  }
}
```

说明：

恢复流程必须重新校验租约、风控状态和会话有效性，不能直接默认恢复。

3.7 关闭会话

请求路径：

```
POST /api/v1/session/close
```

请求参数：

字段	类型	必填	说明
timestamp	int	✅	请求发送时间

session_id	string	✅	当前会话 ID
close_reason	string	✅	结束原因，如 USER_FINISHED
device_summary	object	否	设备侧摘要，例如片段完成情况

请求示例：

```
{
  "timestamp": 1774052100,
  "session_id": "sess_20260321_abcd1234",
  "close_reason": "USER_FINISHED",
  "device_summary": {
    "completed_segment": "story_part_04"
  }
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_close_0001",
  "server_time": 1774052100,
  "data": {
    "session_status": "CLOSED",
    "settlement_status": "PENDING_USAGE_RECONCILIATION"
  }
}
```

说明：

关闭会话时，账本只做临时汇总；最终结算仍应等待平台侧 usage 与回流数据。

3.8 用量回调

请求路径：

```
POST /api/v1/vendor/usage/callback
```

请求参数：

字段	类型	必填	说明
task_id	string	✅	上游任务 ID
session_id	string	✅	业务会话 ID

input_tokens	int	否	输入 token 数
output_tokens	int	否	输出 token 数
rtc_seconds	int	否	媒体时长
event_time	int	✅	事件时间

请求示例：

```
{
  "task_id": "task_20260321_0001",
  "session_id": "sess_20260321_abcd1234",
  "input_tokens": 3421,
  "output_tokens": 8912,
  "rtc_seconds": 518,
  "event_time": 1774052140
}
```

响应示例：

```
{
  "code": 0,
  "message": "success",
  "request_id": "req_usage_cb_0001",
  "server_time": 1774052141,
  "data": {}
}
```

说明：

服务端收到回调后，需把 usage 归因到 user_id、device_id、session_id、lease_id 和 plan_id。

若账本实时估算与平台权威 usage 不一致，应用 settlement_delta 做纠偏。

四、数据模型与账本设计

4.1 核心表

表名	作用
device_registry	设备身份、固件、风险等级、证书映射
user_subscription	套餐、entitlement、周期和 overage 策略
device_binding	用户与设备绑定关系
session	会话生命周期状态
lease	租约、低水位阈值、grace quota

quota_ledger	预算流水账本
vendor_usage_raw	平台 usage 原始数据
usage_settlement	归因与纠偏结果

4.2 流水类型

流水类型	说明
RESERVE	开启会话时预占预算
CONSUME_ESTIMATE	实时估算消耗，仅用于运行期判断
GRACE_CONSUME	grace quota 消耗
RELEASE	未用完预算释放
SETTLEMENT_DELTA	平台权威 usage 回来后的纠偏
MANUAL_ADJUST	人工调整，例如客服补偿或风险止损

4.3 结算规则

推荐顺序：先预占、再运行、后核账、最后结算。

```
会话开始 -> RESERVE
运行期 -> CONSUME_ESTIMATE
平台 usage 回来 -> SETTLEMENT_DELTA
未用完预算 -> RELEASE
人工纠偏 -> MANUAL_ADJUST
```

设备估算仅参与低水位续权；最终扣费必须以平台侧 usage 或账单为准。

五、部署与运维建议

5.1 1000 台接入建议

建议采用单地域、双实例业务服务、托管数据库和托管 Redis 的最小生产配置。媒体面直连云侧，控制面与文本任务面走 APIG / AI Gateway。

层	建议
业务服务	2 台 4 核 8G，主打会话、租约、账本和回调处理
数据库	1 套托管 MySQL，保存 session、lease、账本与审计数据
缓存	1 套托管 Redis，保存 JWT、会话状态、低水位和限

	流状态
网关	优先采用托管 APIG / AI Gateway，不自建统一入口

5.2 10000 台接入建议

建议采用主自建推理接入点 + 备用接入点，业务服务与异步回调处理分离，账本和运维接口独立部署。

层	建议
业务服务	4 台 8 核 16G 或以上，按会话与文本任务面拆分职责
异步与结算	建议独立 worker 节点，专门处理回调、账本和对账
数据库	选择高可用托管 MySQL，并预留读扩展与备份策略
缓存	选择托管 Redis，并单独监控热点键与连接数

5.3 监控与告警

告警项	阈值建议
JWT 登录失败率	5 分钟窗口连续升高时告警
会话开启拒绝率	套餐耗尽与风控拒绝需拆分统计
租约续签失败率	高于基线需排查网关、账本或 entitlement 服务
回调积压	usage 回调延迟超过 10 分钟时告警
账本差异	实时估算与权威计量偏差超过阈值时告警

5.4 版本与变更

本方案建议保留统一的协议版本号和文档版本号。涉及字段新增时，优先采用向后兼容方式扩展；涉及安全与风控切换时，优先通过服务端策略灰度而不是频繁要求设备升级。

文档版本：1.0

最后更新：2026 年 3 月 21 日