AIGW · MULTI-TENANT AI + API GATEWAY

管得住内容，追得到人，算得清账

AIGW 多租户 AI + API 网关：一个控制面、两条数据面，同时治理 AI 调用和普通 HTTP API。

AIGW 多租户 AI + API 网关

让每一次 AI 调用和普通 API 访问都说得清：谁在调用、凭什么放行、用量记在谁名下。通用 API 数据面默认关闭、按需开启；不开启时行为与 v1.9.x 完全一致，AI 链路零改动。

重庆弘创达科技有限公司

可信身份归因　内容安全审核　AI + API 统一治理　调用级核账　私有化部署

出现这些情形，就该给调用建一层入口

多个组织单元、多套应用共用 AI，或已有 HTTP 接口要对外开放时，内容、身份、账与规则的问题会同时出现。

01 · 内容风险要收口 对外——客服机器人、小程序、对外问答的每句输出都代表单位，风险要在出网关前拦下；对内——不当言论与无关用途要能观察、能定位、能留证。

02 · 多部门共用一把 Key 模型侧只看到一个出口，说不清谁在用、更追不到人。

03 · 自建算力池、多上游并存 扩容依据、配额与贡献核算要同一套用量口径；容量与限速各异，要统一入口与连续性机制。

04 · 需要月度核对与审计材料 用量能复核、能锁定、能审计，组织调整后历史数字不重写。

05 · 已有一批 HTTP 接口要对外开放 向兄弟单位、下属机构或合作方给统一入口与可审批订阅，不再各自发 Key、写限流。

章节导览	页码

管得住内容　追得到人　算得清账　业务不中断　一套治理，两类流量

一把 Key，遮住了所有调用者

多个部门、多名员工共用同一套 AI 应用。模型侧看到的，却只是同一个后端出口和同一把 API Key。

人混在一起

谁发起了调用，模型服务不知道；只能看到应用身份，无法落到员工。

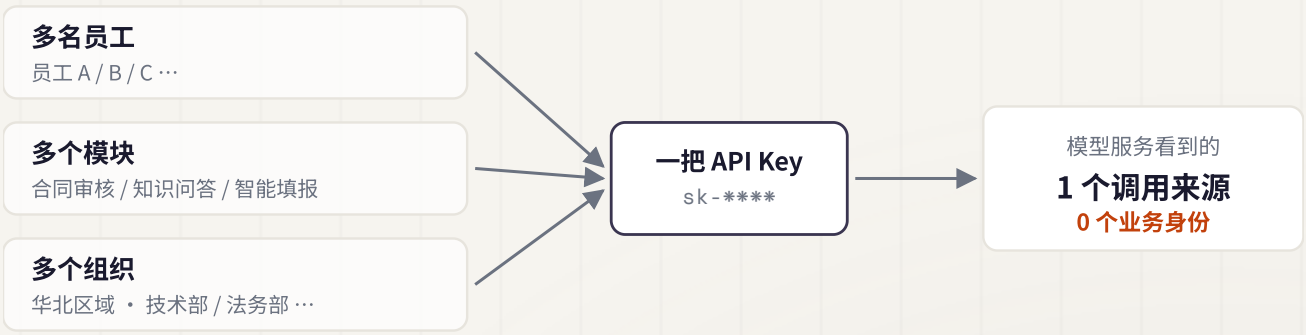
模块混在一起

合同审核、知识问答、智能填报共用出口，用量只能汇总，不能还原到具体业务动作。

组织混在一起

同一平台覆盖多个区域和部门，调用进入模型侧后失去组织归属，账从源头缺少颗粒度。

MANY CALLERS · ONE KEY



人、模块、组织三层信息在出口处被抹平——不是没有记录，是记录里没有业务身份。

问题不在“有没有调用记录”，而在记录里有没有业务身份。

没有身份，内容出了问题也追不到人；用量涨了也说不清是谁在涨。

一把 Key　多人共用　模块混用　归属缺失

看得见卡在忙，看不见替谁忙

“调度”在不同层级解决不同问题。资源层的指标再完整，也不能自然回答业务归属。

FIVE LAYERS · WHO ANSWERS WHAT

企业治理层	租户、部门、应用、Key、计量账目	AIGW
推理请求层	每个请求选择哪个模型副本	推理网关 / Router
模型服务层	模型实例、副本与扩缩容	模型服务平台
作业准入层	训练作业、团队 GPU 配额	作业队列
GPU 设备层	GPU、显存、切分与分配	容器编排调度器

越往上越贴近业务：这笔消耗记在谁名下，只有最上层回答得了。

各层各管一件事

GPU 设备层决定显存怎么切，作业准入层决定作业能不能排上，模型服务层决定模型实例怎么伸缩，推理请求层决定请求进哪个副本。这些层都不回答“这笔消耗记在谁名下”，也都不看内容。

企业治理层的问题

哪个区域、哪个部门在使用；哪个员工、哪个应用发起；哪个模块、哪个模型产生用量；这次请求与回答是否合规。

边界必须说清

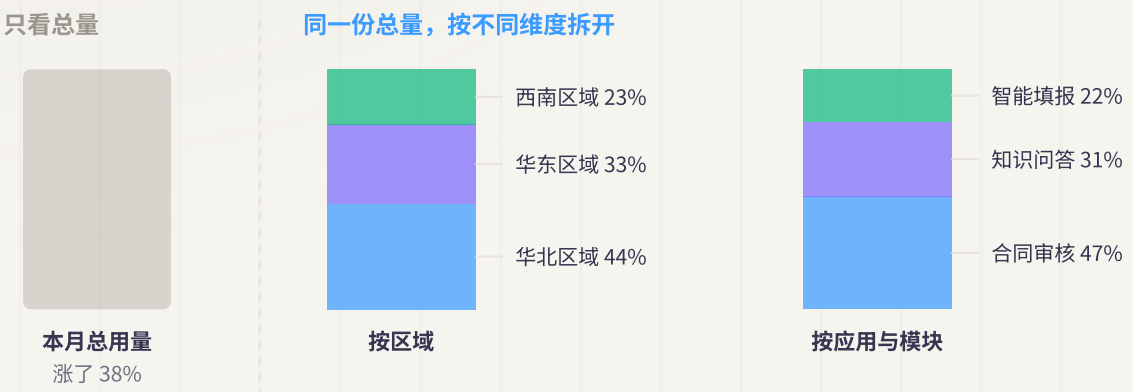
网关记应用层的账、管应用层的内容，调度平台管资源层的卡。两类数据各有职责，拼在一起，才形成从业务调用到算力资源的完整视图。

资源视角　业务视角　应用层账　资源层卡

共建之后，扩容、配额与贡献怎么算

共建方关心的不是简单收费，而是“用量说得清”。账目颗粒度不足，管理动作就缺少共同依据。

SAME TOTAL · DIFFERENT LEDGERS



扩容找谁 · 配额给谁 · 贡献怎么算——三个问题都要先有右边这两本拆开的账。

扩容找谁

当资源持续紧张，管理者需要知道增长来自哪个区域、部门、应用与模块，而不是只看到总量上涨。

配额给谁

当多个应用争用同一算力池，配额应建立在可核验的历史用量与业务优先级上。

贡献怎么算

共建投入与实际使用需要同一套计量口径，按模型差异折算，按组织与应用汇总。

算力可以共建，账必须分明。

没账就没公平。先让每次调用有归属，再讨论共建规则。

共建分账　扩容依据　配额依据　贡献核算

一个控制面、两条数据面，把治理收口在同一层

AI 数据面承载模型调用，通用 API 数据面承载普通 HTTP 服务；两者共用身份、RBAC、审计与集群运维。

ONE GATEWAY · FOUR JOBS

上下游职责不变，
网关只加一层治理与账。

一条调用，落成一条责任链

两条数据面，各管一类流量

AI 数据面从身份、策略、内容、路由到 Token 计量形成责任链；通用 API 数据面从服务目录、订阅准入、策略生效到字节与时延访问账形成治理链。两本账分表、口径不混。

升级边界写在前面

通用 API 数据面默认关闭、按需开启，不打开时行为与 v1.9.x 完全一致；AI 数据面在四批升级中零改动，每批都复跑 /v1 golden 回归用例。主版本进位到 2.0 源于新增通用 HTTP 代理入口带来的部署形态变化，不是重写。

上下游职责不变

模型服务以兼容端点承接请求；算力调度平台继续负责实例伸缩与利用率管理。

最终形成：每次 AI 调用与 API 访问，都能从“谁在用”追到“凭什么放行”，再追到“如何落账”。

一个控制面　两条数据面　身份链　内容审核　访问明细

把身份、内容、用量与 API 治理放在同一条链上

关键不是多一个功能，而是 AI 调用与普通 API 共用身份、订阅、策略与审计，不再各自维护一套治理链。

FROM METRIC TO LEDGER

用量作为指标

- 看趋势与总量
- 口径随当前配置变化
- 组织调整后数字会被重写
- 用于监控与预算参考

用量作为账目

- **可归属** 组织 / 应用 / 模块 / 用户 / 模型 同时落账
- **可冻结** 落账即写组织快照，人事变动不回写
- **可核对** 折算逐笔可复核，分组之和等于汇总
- **可锁定** 账期锁定不再重算，锁账动作进审计

为什么不是“一张更好的看板”

指标可以随时重算，账目必须能被复核和锁定；过滤只回答“这句话该不该发出去”，追责还要回答“这句话是谁发的、在哪个应用的哪个模块里发的”。两者对数据的要求根本不同：账目与证据都要求归属在落账那一刻就固定下来，此后不被任何配置变更改写。

这套组合意味着什么

企业组织树、调用级多维归因、组织快照冻结、月度锁账、内容命中记录、容量感知与排队，以及 API 资产版本、套餐订阅、五层策略和访问账，共同构成一条从人到内容、从准入到账目都能对上的责任链。

启用方式不改变既有 AI 链路

通用 API 数据面默认关闭、按需开启；不打开时行为与 v1.9.x 完全一致，AI 链路零改动。

可归属　可追责　可核对　一套治理，两类流量

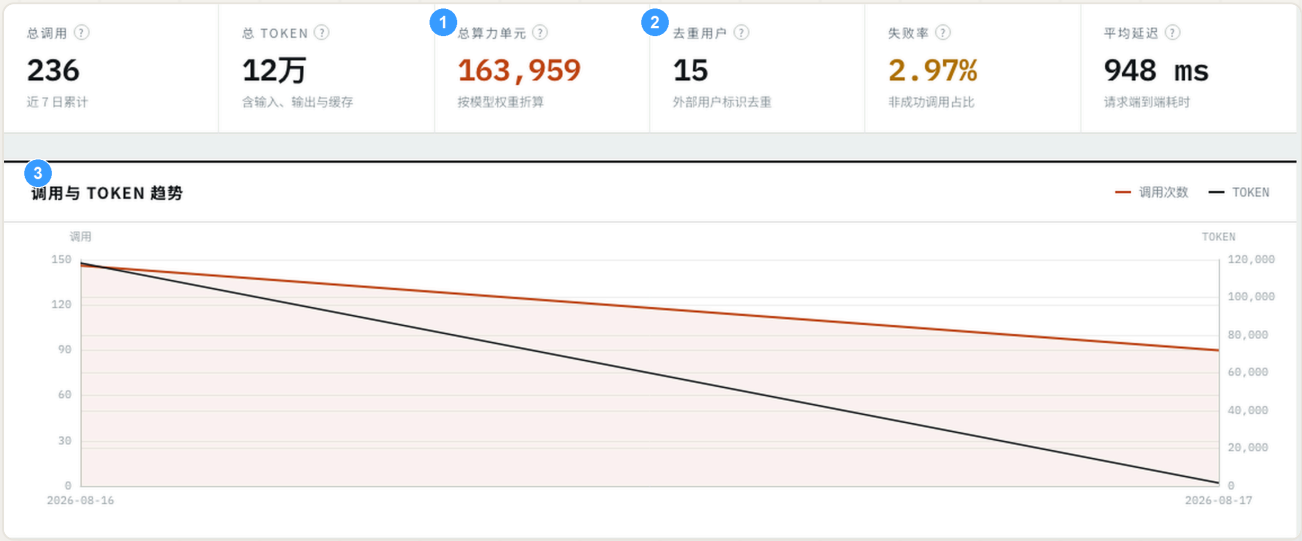
把混在一起的调用，拆成说得清的账

计量不只数调用次数。AIGW 把身份、组织、应用、模块、模型与 token 放进同一条用量记录。

本章回答四件事

- 系统账、用户账、部门账如何逐层形成
- 六个业务维度如何组合查询
- 不同模型如何折算为共同计量口径
- 人员与组织变化后，历史账如何保持原貌

从总量走向归属，从归属走向可核验。



看板给的是态势，不是账：先看总量与趋势，再进报表按维度拆开。

① 按模型档位权重折算的算力单元　② 去重后的终端用户数　③ 近 7 日调用与 TOKEN 双轴趋势（演示数据）

三本账　六维报表　算力单元　快照冻结

先有系统账，再细到用户账与部门账

不同接入条件对应不同归因颗粒度。AIGW 让每一步独立见效，也让账本可以持续变细。

THREE LEDGERS · STEP BY STEP

每一步独立见效：只做第一步就已经有应用与模块的账，接入越完整账本越细。

第一本账不需要改代码

只要按应用或模块分别签发 Key，应用与模块维度的账当天就有；业务系统的调用代码一行不动。

第二本账要传对标识

用户标识必须是稳定的员工编号或不变的用户 ID，不要传姓名、不要传部门名称——姓名会重、部门会改，都不能作为归账依据。

第三本账靠组织映射的公信力

组织映射与人事口径对齐之后，部门账才站得住；映射只维护在网关一处，避免多个系统各自维护一套组织。

系统账 用户账 部门账 逐层接入

同一份用量，按管理问题重新组合

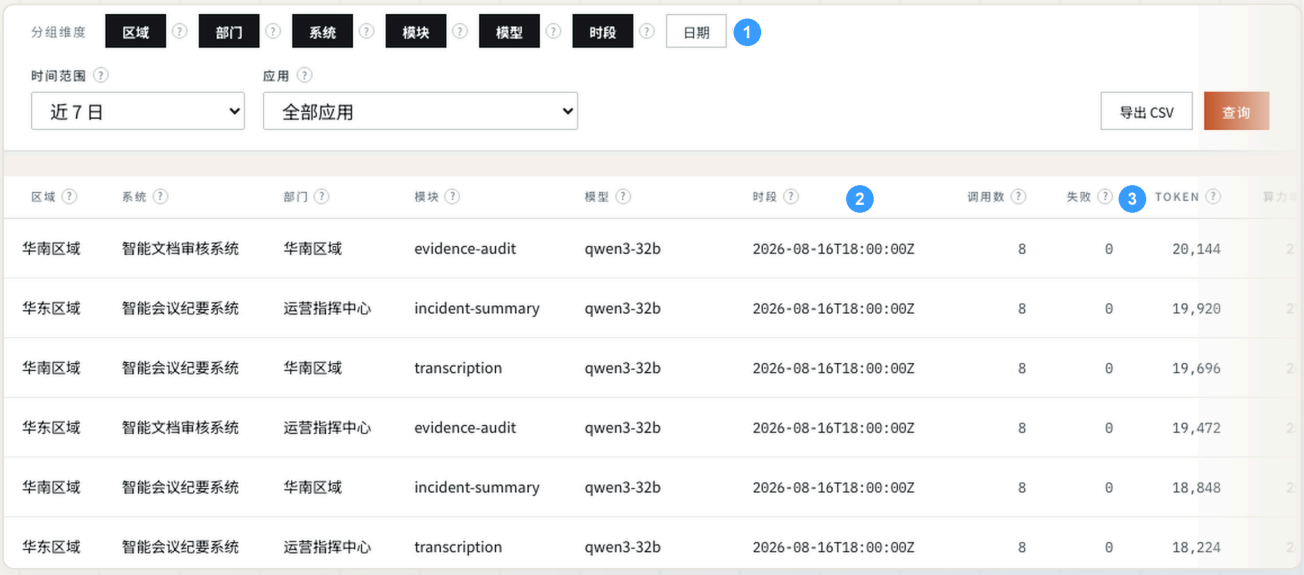
区域、部门、应用（业务系统）、模块、模型、时段可任意组合，并可从汇总钻取到调用明细。

六个维度

- 区域：不同区域分别用了多少
- 部门：组织内部用量如何分布
- 应用：哪套业务系统产生调用
- 模块：具体业务场景的用量
- 模型：请求与实际执行模型如何分布
- 时段：用量在小时、日期与账期内如何变化

典型管理问题

- 部门 × 模型：哪些部门在使用高档位模型，是否符合业务定位
- 用户 × 模块：谁在大量调用、用在什么场景，是否与本职工作相符



同一份用量，勾选不同维度就是不同的账；每一组还能继续钻取到调用明细。

① 六个分组维度任意勾选 ② 勾中的维度成为分组列 ③ 每组的调用数、失败数与 TOKEN（表格向右还有算力单元与操作列）

任意组合　明细钻取　小时汇总　月度锁账

不是只数次数，而是按 token 与模型档位折算

一次长内容生成与一次短问答，不应被记成相同用量；不同模型的资源消耗，也需要共同口径。

WEIGHTED UNIT · ONE FORMULA

同一套调用明细，由租户 billing_mode 决定用哪一轨。

折算过程可以逐笔复核

权重来自模型档位配置，由参与方按统一规则确定。每笔明细都保留输入、输出、缓存 token 与请求模型、实际模型，任何一笔的折算结果都能自己算一遍核对；月度锁账时权重版本一并留存，事后追溯用的是当时的权重，不是现在的权重。

为什么共建场景不谈金额

共建方之间要解决的是“谁用了多少、谁该分摊多少”，一旦换算成钱，就要先谈定价，反而把账吵得更复杂。加权算力单元只回答用量口径这一件事；确实需要计价时，租户改用 `billing_mode` 的金额口径，调用明细不用重采。

token 计量　模型权重　双轨口径　逐笔复核

人会调动，机构会调整，历史账纹丝不动

报表不能用今天的组织关系重写昨天的归属。AIGW 在调用落账时，把当时的组织路径一并冻结。

落账当时

用户标识经组织目录解析，区域、部门、组织路径、应用名称与调用用量共同写入明细快照。

组织调整后

员工调岗、部门更名或组织层级调整，只影响后续调用的归属；既有账期仍读取落账时快照。

管理结果

- 上月已经确认的部门账不随现状变化
- 新调用按新的组织关系继续归因
- 月度锁账后，补算与重算不改变已锁定期间数字

组织快照不是展示字段，而是账目与证据的公信力基础——一条半年前的命中记录，查出来还是当时那个部门。

部门	调用数	失败	TOKEN	算力单元	均延迟	操作
华南区域	24	0	58,688	79,568	1,088 ms	明细
运营指挥中心	24	0	57,616	77,896	1,071 ms	明细
财务与资产部 ①	110	5	2,033	3,639 ②	1,544 ms	明细
法务部	40	2	760	1,428	26 ms	明细
技术部	38	0	760	1,428	27 ms	明细

表格分页展示，合计口径以上方汇总卡为准（全量统计）

计量口径：算力单元 · 权重版本 49 · 数据截止 2026/8/17 12:35:01 ③ 上一页 下一页

部门改名之后，同一时段的部门账数字与改名前一致——报表读的是落账时的组织快照。

① 改名后的新部门名称 ② 同一账期的算力单元未变 ③ 计量口径、权重版本与数据截止时间随报表一并给出

落账即冻结 历史不重写 新旧分离 锁账可审

不是自报家门，是签名验身

网关能看见请求，却不能凭空知道“这次调用替谁发起”。可信身份必须由掌握登录态的业务系统带入，并由网关验证。

身份链的基本原则

- 业务系统掌握当前登录员工与模块
- 调用方把稳定标识放入 AI 请求
- 需要可信归账时，由服务端签发调用方身份令牌（X-Caller-Token，简称 L3 令牌）
- 网关验签后完成组织映射、策略判断、内容审核与落账

可信身份不是多记一个字段，而是让“谁在调用”成为可验证的输入。

TRUSTED IDENTITY FIRST

身份与组织两段确立，后面四段才有依据

L1 / L2 是未签名输入，用于统计归因；需要可信归账时由 L3 令牌承担——
网关不替代企业统一认证中心，它只验证「这次调用替谁发起」这一件事。

调用方身份　服务端签名　网关验签　可信归账

一次调用，八个问题都能回到同一条链上

从用户、组织到结果与追踪，AIGW 把身份、授权、资源和计量统一到调用上下文中。

身份链问题	可管理结果
WHO · 谁调用	识别发起调用的员工编号或稳定 ID
ORG · 属于哪里	解析员工所属区域与部门，落账时冻结组织路径
AGENT · 经由什么	识别应用、Agent 与模块
WHY · 凭什么放行	根据策略判断准入，并记录授权结果
WHAT · 用了什么	记录请求模型、实际模型与上游服务
HOW MUCH · 用了多少	记录输入、输出、缓存 token，并形成算力单元或金额结果
RESULT · 结果如何	记录成功或失败状态与错误码
TRACE · 如何追踪	以请求 ID 贯穿调用、计量、命中记录与审计链路

这八个问题共同回答：谁在什么组织、通过什么应用、基于什么规则、使用了什么资源，结果如何核验。内容命中记录挂在同一条链上，所以“这句话是谁发的”也有答案。

八个问题不是八张报表，是同一条调用记录里的八组字段。

① 请求 ID 贯穿调用、计量与审计（TRACE） ② 落到具体员工标识（WHO） ③ 模块标识（AGENT）——同一条记录向右还有实际模型、状态码与用量

WHO ORG AGENT RESULT TRACE

按接入能力选择档位，按可信要求决定边界

身份信息可以逐步接入，但统计归因与可信归账必须明确区分。平台能力越完整，身份链越细、验证强度越高。

档位	身份机制	归因能力	可信边界
L1	OpenAI 标准请求体 user 字段	用户级归因	字段由请求方填写，不具备签名验证
L2	X-Caller-User / X-Caller-Module 请求头	用户 + Agent / 模块归因	请求头由调用方填写，不具备签名验证
L3	X-Caller-Token，HS256 签名 JWT	用户 + 组织 + Agent/模块 + 计量归账	网关验签，适用于可信归账

PRECEDENCE & TRUST BOUNDARY

用户标识取值优先级：L3 令牌 → 请求头 → 请求体

多种身份并存时

用户标识按 L3 令牌、请求头、请求体的顺序取值；Agent 标识优先采用已验证 L3 令牌中的信息。

需要可信归账时

管理员可在策略中要求有效 L3 身份；未通过验证的请求不进入受保护的模型调用链。涉及追责的场景（如面向公众的应用）建议直接要求 L3。

能力边界 提供统一身份接入、授权、调用归因与审计，衔接企业目录（LDAP / SCIM）；不替代企业统一认证中心。

L1 兼容　L2 扩展　L3 验签　分级接入

终端用户身份不会自然到达模型网关

AI 编排平台承接工作流与应用运行，但其下游模型请求通常代表平台自身，而不是直接代表终端员工。

实测对象	实测版本	技术事实
Dify	1.16	不透传终端用户身份
FastGPT	v4.15.7 + aiproxy v0.6.5	不透传终端用户身份
Coze	开源版（2026-08 实测）	不透传终端用户身份

上述三项结论均为 2026-08 实测，版本范围为 Dify 1.16、FastGPT v4.15.7 + aiproxy v0.6.5、Coze 开源版，只陈述身份透传现状，不评价产品优劣。

WHERE IDENTITY STOPS

这意味着什么

若需要按员工、按部门形成可信用量账，或需要把一条内容命中追到具体经办人，终端身份必须由掌握登录态的调用方显式带入，再由 AI 网关完成验证、组织映射与落账。

接入不改变平台职责。

编排平台继续管理应用与工作流；AIGW 负责下游出口的身份归因、内容审核、策略准入、模型路由与用量审计。

实测版本　身份透传　职责边界　网关归因

不知道是谁在说话，就管不住说了什么

算力是私有的，但服务对象可能在外面。内容风险有两个面，两个面的处置方式并不相同。

TWO SURFACES · ONE PIPELINE

同一个网关里，对外应用设拦截、对内应用设观察，按应用分别下发，互不影响。

命中记录带完整归因，但只保留脱敏摘要与内容哈希，不留存原文——追得到人，也停在该停的地方。

对外服务：输出即代表单位

客服机器人、微信小程序、对外问答等应用，每一句回答都以单位名义发出。不当内容一旦发出就是舆情与合规事件，必须在出网关之前拦下，而不是事后删除。

对内使用：要能追责，不必窥私

不当言论、贬损单位与他人的内容、把 AI 用在与业务无关的用途，需要能定位到人、能留下证据、能形成处理依据。AIGW 的取舍是：命中记录带完整归因，但**当前留存策略只保留脱敏摘要与内容摘要哈希，不留存原文**；连“谁查看了命中详情”这个动作本身也进审计。

本章回答三件事

- 审核流水线怎么编排，关闭 / 观察 / 拦截三种模式各用在什么场景
- 词库如何自主维护，命中判定如何分级
- 命中之后留下什么，追责到什么程度为止

对外拦截　对内留痕　脱敏摘要　审计闭环

先用词库快筛，再交模型复核；先观察，再拦截

审核既要拦得住，也不能把业务拖慢、更不能一上线就误伤。所以它被拆成多阶段流水线和三种运行模式。

三种运行模式

总开关一处切换：**关闭**（不做审核）、**观察**（异步审核并记录命中，不影响业务请求）、**拦截**（按流水线阶段与处置规则执行）。新规则先在观察模式跑一段，确认误伤率之后再切拦截。

流水线怎么编排

按从上到下顺序执行，最多 8 个阶段，可自由组合关键词阶段与模型阶段：先用本地词库做低成本快筛，再把可疑内容交模型复核。每个阶段单独选择命中处置——观察、交下一阶段复核、拦截、拦截并短路；关键词阶段还可设最低命中分，避免单个弱命中就拦。

词库自主维护

按分类分库维护敏感词，支持搜索、导入导出与批量粘贴（可设分隔符与冲突策略），单库可承载数万条；既有词表直接导入，不必迁就产品预置。审核配置按应用分别下发——对外应用设拦截、对内应用设观察，同一个网关里并存。

总开关三种模式 + 多阶段流水线：新规则先在观察模式跑，确认误伤率后再切拦截。

① 关闭 / 观察 / 拦截，按应用分别下发 ② 流水线按序执行，最多 8 个阶段，可加关键词阶段与模型阶段 ③ 关键词阶段选择参与词库与最低命中分

关闭 / 观察 / 拦截　多阶段流水线　本地词库　模型复核

命中要追得到人，也要停在该停的地方

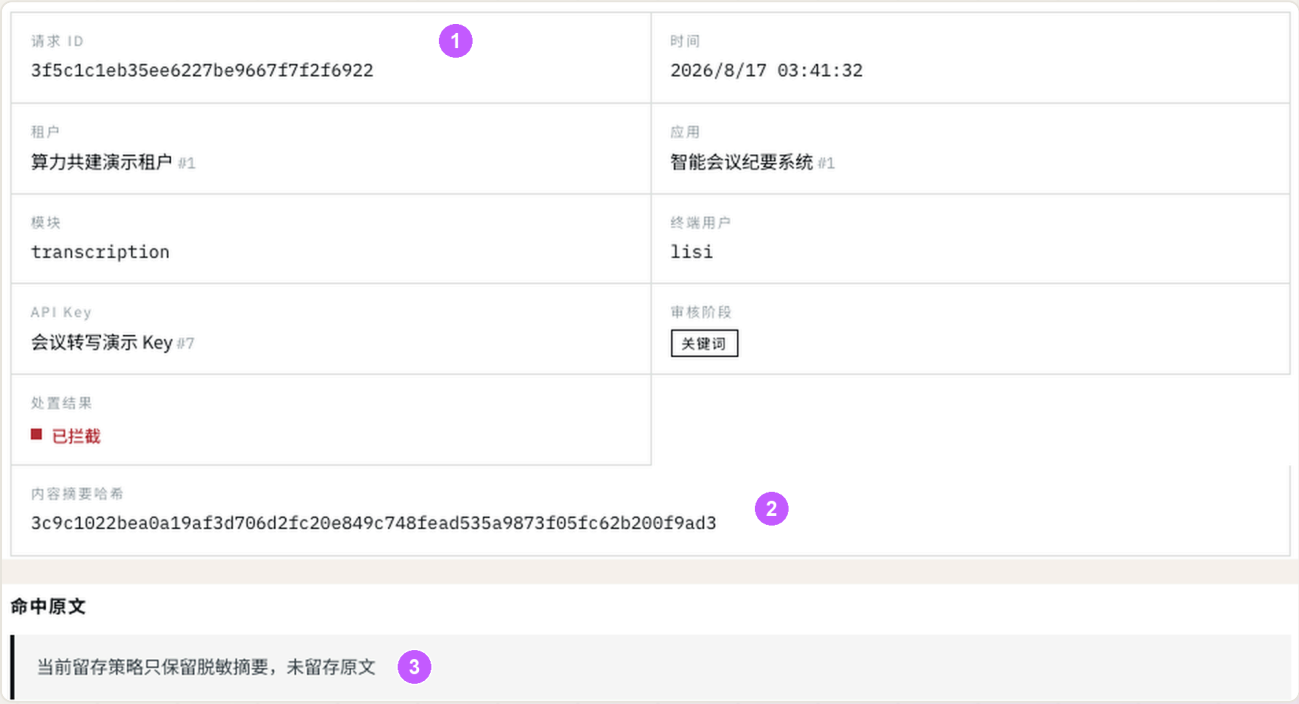
命中记录落不到具体的人与模块，就形不成处理依据；连原文都留着，又会变成新的风险源。

留证到哪里为止

命中记录带完整归因，可按请求 ID 直接查到那一次调用；L3 验签过的调用能落到具体员工与模块。但命中详情按**当前留存策略只保留脱敏摘要与内容摘要哈希，不留存原文**——哈希用于核对“是不是同一段内容”，不用于还原内容。查看命中详情的动作本身写入审计：管理员看了什么，同样有记录。

拦截了也照常记账，异常了也不阻断业务

拦截模式下命中请求返回 400，用量与命中记录仍按规则落库，不会因为被拦就在账上消失；审核服务异常时流水线超时放行，不阻断模型业务，运行状态由监控指标与“审核激增”告警暴露。内容安全结果用于辅助判断，不替代人工决策。



一条命中记录能追到人与模块；按当前留存策略只保留脱敏摘要与内容摘要哈希，未留存原文。

① 完整归因：请求 ID、租户、应用、模块、终端用户、API Key、审核阶段与处置结果 ② 内容摘要哈希（只用于核对是否同一段内容） ③ 图中环境的留存策略：只保留脱敏摘要，未留存原文

完整归因　脱敏摘要　查看即留痕　超时放行

让上游容量变化，不直接变成业务中断

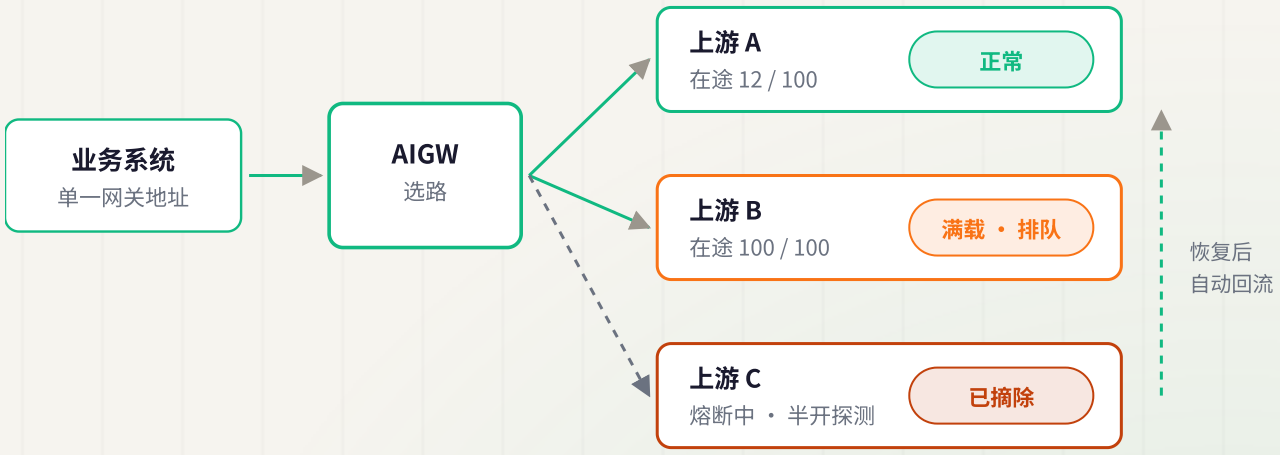
多上游不是简单轮询。AIGW 综合容量、并发、速率、权重与健康状态选路，并为满载与变更预留过渡机制。

本章关注三类变化

- 上游故障与恢复：熔断、半开探测、自动回流
- 算力满载：排队等待、超时背压
- 日常运维变更：换 Key、换模型、升级节点

业务系统只面对一个网关地址，变化在网关层被识别、调度和记录。

ONE ENTRY · THREE UPSTREAM STATES



容量感知　熔断回流　满载排队　变更连续性

不只看“能不能连”，还看“现在接不接得住”

上游服务能力不同、负载持续变化。AIGW 将静态配置与运行状态一起纳入选路。

容量感知选路

- 每个上游配置权重、最大在途并发、RPM 与 TPM 容量
- 采用 least-in-flight 策略：优先选择当前在途请求更少的可用上游
- 同层候选已满时继续选择其他候选；没有可用候选时按路由层级处理

故障隔离与恢复

- 连续异常触发熔断，暂时摘除问题上游
- 业务流量转向仍可用的上游服务
- 上游恢复后进入半开探测，通过后自动回流

管理者得到的不是一张静态路由表，而是一套随容量与健康状态变化的调度结果。

左侧上游已被熔断摘除，业务流量转向右侧仍可用的上游。

① 熔断中（右侧为接流正常） ② 在途 / 上限、RPM、TPM 实时可见 ③ 熔断阈值与半开探测参数逐上游可配

多上游 least-in-flight 熔断半开 自动回流

算力满载时，先排队，不把压力直接推给模型服务

GPU 是稀缺资源。超过上游并发容量的请求，立即失败会放大业务抖动，也会掩盖真实容量需求。

QUEUE INSTEAD OF FAIL FAST

满载时的处理顺序

1. AIGW 先在同层路由中寻找仍有容量的上游。
2. 当前层没有可用容量时，按配置继续评估后续候选。
3. 全部候选满载时，请求进入等待队列，而不是立即报错。
4. 等到容量释放后继续转发；超过队列窗口则返回 429，并携带 `Retry-After`。

为什么这是关键行为

它把短时拥塞与持续容量不足区分开：前者通过等待吸收，后者通过明确背压暴露，便于应用退避重试，也为后续扩容提供依据。

容量队列　429 背压　退避重试　扩容依据

换钥匙、换模型、升级节点，都给业务留出过渡空间

真正影响连续性的，往往不是日常调用，而是必须发生的变更。AIGW 把变更过程设计进运行机制。

Key 轮转双活

管理员签发新 Key 后，新旧 Key 在设定的宽限窗口内并行有效；业务系统完成切换后，旧 Key 再退出使用，避免换钥匙形成硬切窗口。

模型别名灰度

业务继续请求既有逻辑模型名，网关按应用或上游范围映射到实际模型；先在局部观察，再逐步扩大，异常时撤回映射即可回退。

节点排空

升级前先将节点移出新请求入口，待在途请求完成并冲刷用量后再重启回流；负载均衡动态解析节点，仅在请求尚未开始处理时，对排空 503 或连接失败重试 POST，响应开始下发后不再重试。

集群档演练实测

集群档演练条件：k6 恒定 30 并发，经 nginx 发送非流式 POST，任何非 200 均计为失败。双节点滚动升级持续 6 分钟，共 46,389 请求、0 失败；控制面停机 60 秒，共 7,638 请求、0 失败。

时间	请求 ID	应用	用户标识	模块	实际模型	响应模型	状态码	TOKEN	算力单元	延迟
2026/8/17 02:41:33	b6dbb2d7f357b94f3984d d238f8f96bf	智能会议纪要系统	zhangsan ①	transcription	qwen3-32b	—	200	19	33 ③	28 ms
2026/8/17 02:40:50	f9399ef387cc7933545d2 1df6145c05c	智能会议纪要系统	zhangsan	transcription	qwen3-235b- a22b ②	—	200	19	33	29 ms
2026/8/17 02:39:06	073f020621a6afe864ccd 1af796e2d82	智能会议纪要系统	zhangsan	transcription	qwen3-235b- a22b	—	200	19	33	20,012 ms
2026/8/17 02:39:06	f8bbbe29fa4acb8a144e3 039f2081b0b	智能会议纪要系统	zhangsan	transcription	qwen3-235b- a22b	—	200	19	33	20,013 ms
2026/8/17 02:39:06	2a638b2c39884110295c6 27cfab56d0a	智能会议纪要系统	zhangsan	transcription	qwen3-235b- a22b	—	200	19	33	20,015 ms
2026/8/17 02:39:06	43dff5ebcc3268a98f46b d7784c641c5	智能会议纪要系统	zhangsan	transcription	qwen3-235b- a22b	—	200	19	33	20,015 ms

灰度只对指定应用生效：业务继续请求同一个逻辑模型名，实际执行模型已经换新。

① 调用落到具体员工标识　② 实际执行模型已是灰度后的新模型　③ 每笔的 TOKEN、算力单元与端到端延迟

Key 双活　模型灰度　节点排空　演练实测

规则在入口收口，过程在一处留痕

租户、应用与 Key 的要求不同，但进入模型服务之前，需要形成唯一、明确、可审计的生效规则。

FIVE CONTROLS · ONE ENTRY

五类要求分散在不同系统里就会各说各话；收到一处，才有唯一、明确、可审计的口径。

一层入口，收口五类治理对象

谁可以调用（身份、Key 状态、来源范围）；可以调用什么（模型白名单与拒绝规则）；可以调用多少（配额、RPM、TPM、并发）；内容如何处理（审核流水线与运行模式，见第三章）；操作如何追溯（调用日志、命中记录、审计日志与月度锁账）。

审计链闭合

调用明细记录请求 ID、身份、模型、状态与用量；命中记录说明审核结果与处置；审计日志记录 Key 吊销与轮转、策略变更、配置修改、命中详情查看与锁账等敏感操作——谁、什么时候、对哪个对象、为什么，中文详情可直接阅读，原始值仍可用于技术核对。

月度锁账

账期一经锁定不再重算，锁账时权重版本一并留存；补算与重算不改变已锁定期间的数字，锁账动作本身进审计。

统一准入　配额限流　审计留痕　月度锁账

租户定边界，应用定场景，Key 定到调用入口

策略按租户、应用、API Key 三级继承折叠。越具体的层级补充自身要求，最终形成一份生效结果。

TENANT · APP · KEY

折叠规则

- **限额**
  取更严的那个值
- **白名单**
  取各层共同允许的范围
- **可信身份**
  任一层要求 L3，即必须验签
- **未设置的字段**
  沿用上层，不被覆盖
- **生效结果**
  每把 Key 都能单独预览核对

为什么要分三级，而不是一把规则管到底

治理边界由租户方定，业务场景由应用负责人定，具体入口由运维在签发 Key 时定——三方关心的粒度不同，写在一起就会互相覆盖。分级之后，每一层只维护自己那部分，没设置的字段沿用上层；最终生效结果由网关折叠计算，上线前可逐把 Key 预览核对。

租户策略　应用策略　Key 策略　生效预览

先按可用性选形态，再按在途并发定节点

AIGW 以同一个二进制支持单机、双机主备与集群部署，面向受控私有化交付。上集群的核心理由是可用性与滚动升级，而不是单纯追求进程数量。

SINGLE · ACTIVE-STANDBY · CLUSTER

单机	双机主备	集群
现场 POC / 单区域试点	常规生产	多区域共建 / 滚动升级
AIGW 二进制 ×1	AIGW ×2 + 前置 VIP	数据面节点横向扩展
PostgreSQL	共用 PostgreSQL	配置与账务集中存放
本机计数	Redis 统一计数	未落库用量本地 WAL

同一个二进制，三档形态；上集群的理由是可用性与滚动升级，不是进程数量。

容量测算基线

以 2 万注册用户的政企系统测算：日活约 6,000，日调用约 12 万次，峰值 QPS 25—40，在途并发 200—300，用量明细约 12 万行/天、360 万行/月。单数据面节点稳妥承载约 500 在途连接，3 节点覆盖该峰值的 5 倍余量。

单机 all 模式实测

单机 all 模式、macOS arm64，k6 v2.2.0；300 并发稳态 60 秒，随后 600 并发尖峰，模拟上游注入 120ms。共 38,633 请求，失败率 0.00%，0 个 5xx；p95 为 1361ms（含模拟上游注入的 120ms）；新增落库 38,633 条，与发出请求数一致。

v2.0.00 工程实证

Go 源码 76,765 行，486 个测试函数；Playwright 运行时展开 145 条 UI 用例，覆盖全部运营台页面；数据库迁移实际 33 个。

单机试点　双机主备　集群部署　容量测算

一套治理，两类流量

AI 调用和普通 HTTP API 走不同的数据面，共用同一个控制面的身份、订阅、策略、RBAC 与审计。

本章回答三件事

- 两条数据面怎样分工，为什么两本账不混口径
- API 资产怎样从登记、发布到稳定可调用
- 套餐订阅与五层策略怎样得出可解释的生效结果

开关边界

通用 API 数据面由 `AIGW_API_CATALOG_ENABLED` / `AIGW_API_PLANE_ENABLED` 控制，默认关闭、按需开启；不打开时行为与 v1.9.x 完全一致，AI 链路零改动。

ONE GOVERNANCE · TWO PLANES

一套治理，两类流量。通用 API 数据面由两个开关控制，默认关闭。

一个控制面　两条数据面　默认关闭　按需开启

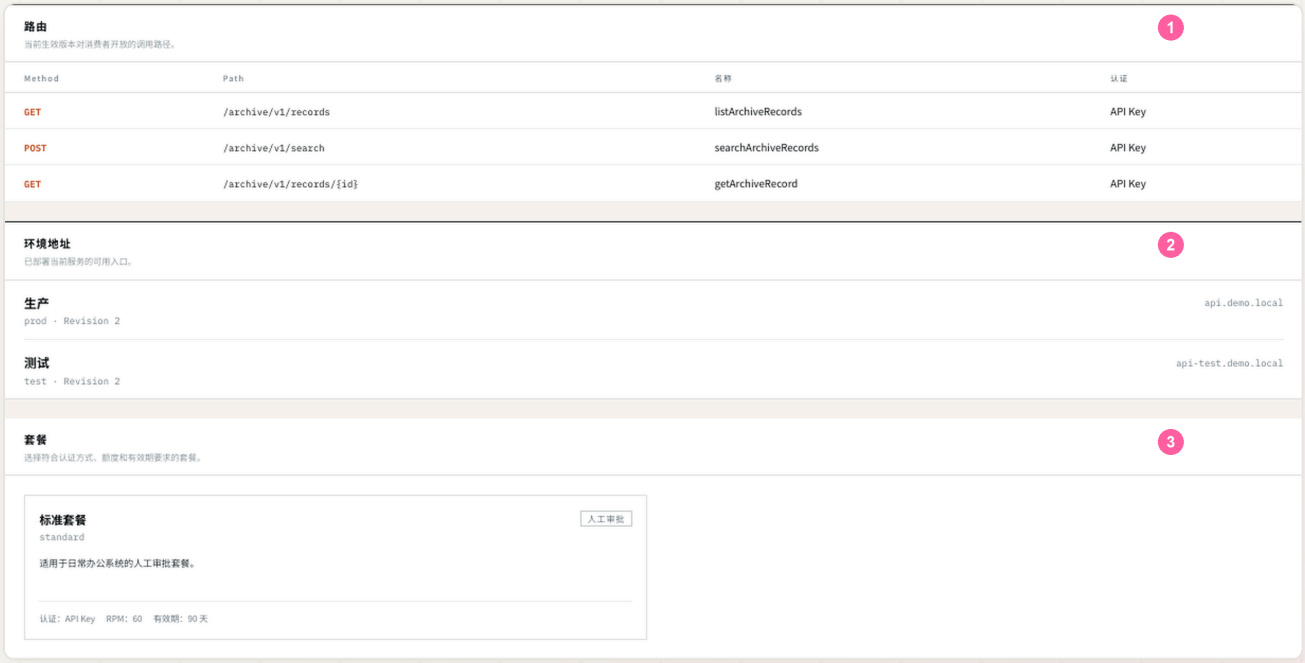
分开承载，共用治理

两条数据面各承载一类流量，共用同一套身份、审计与运维。

	AI 数据面	通用 API 数据面
承载	/v1/* 九个 AI 端点	/v1 之外按发布配置匹配
特有能力	模型别名、协议转换、内容安全、容量感知路由	路由匹配、后端池、SSRF 防护、订阅准入、策略
计量	Token 账	字节与时延账（分表）

目录只对外露该露的

只展示已发布与已弃用的服务，不暴露后端地址与后端池；报表不做跨口径合计。



目录只展示已发布与已弃用的服务，调用方看不到后端地址与后端池配置。

① 服务名称与状态 ② 可订阅套餐 ③ 对外调用说明（演示数据）

实验室性能实测

k6 300 VUs / 60s、GET、同机 8 核：裸后端 74,808 req/s，经网关匿名 22,008、API Key 17,404 req/s，三组均 0 失败。p95 反映同机 CPU 饱和排队，不是网关处理成本；进程内基准单请求净开销约 1.5 µs。

共用治理　分表计量　服务目录　实验室实测

配置发布后不改历史，每次变更都有新记录

把 API 作为可治理资产：登记服务、编辑草稿、发布到指定环境，再由数据面稳定执行已发布配置。

不可变版本

版本发布后不可修改；变更与回滚都以新草稿、新发布记录完成。

发布前校验

发布时检查保留路径、路由、后端池、后端地址与路径模板；后端地址每次拨号重新校验，内网后端须显式允许。

IMMUTABLE VERSION · APPEND-ONLY RELEASE

配置版本一经发布不可修改；回滚是新动作，不是撤销或改写旧记录。

一致匹配与热加载

路由按 Host、路径、方法与 Header 匹配，以固定优先级保证各节点结果一致。数据面每 5 秒比对版本，预编译后原子切换；在途请求不受影响，失败则保留上一版并告警。

不可变版本　环境发布　原子切换　访问账

准入有审批，限制有解释

目录负责告诉调用方“有什么”，套餐与订阅决定“能不能调”，策略引擎计算“能调多少、按什么规则调”。

从申请到生效

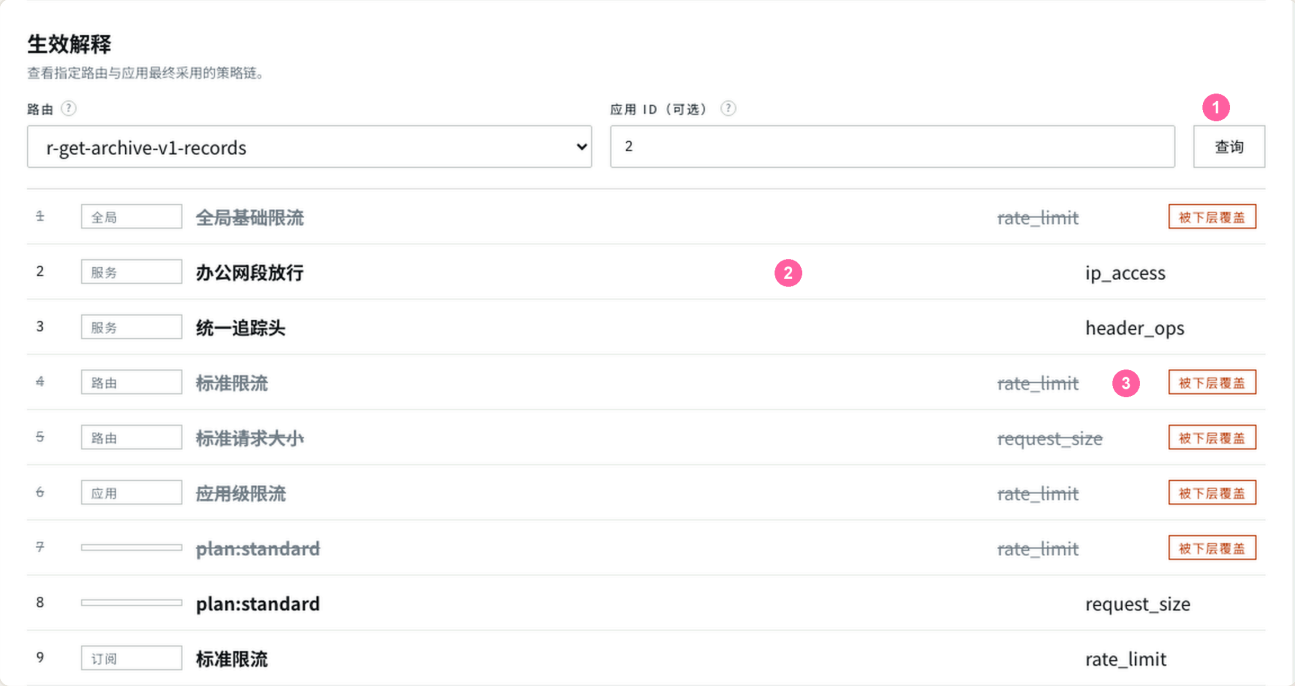
订阅按申请 → 审批 → 生效 → 到期提醒闭环管理；审批防并发、可安全重试。服务配置套餐后，无生效订阅的 Key 返回 403 `not_subscribed`；本版本到期通知只落库、不发送。

认证边界

数据面要求同租户 API Key，也可验证外部 IdP 的 JWT。AIGW 只验断言；MFA / SAML / 密码策略由 IdP 承担，运营台令牌不能用于数据面。

策略不只告诉你“有哪些”

策略覆盖 IP 访问、频次 / 并发 / 日额度、请求大小与 Header 变换，可绑定在全局 → 服务 → 路由 → 应用 → 订阅五层，同类型后层覆盖前层。“生效解释”列出最终策略及被覆盖项。



生效解释逐层列出最终策略，并标明哪些配置已被后层覆盖。

① 策略来源层级 ② 最终生效项 ③ 被覆盖项与原因（演示数据）

套餐订阅 审批生效 五层策略 生效解释

说清楚适合谁，
也说清楚现阶段不适合谁

把边界写在前面，能省掉双方大量的售前解释与错位比较。

适合

- 政企、集团与大型组织，需要私有化部署
- 有面向公众的 AI 应用，内容风险必须在出网关前收口
- 多区域、多部门、多应用共同使用 AI，需要追到人与模块
- 自建模型服务、共建算力池或多个模型供应来源并存
- 已有 HTTP 接口需向兄弟单位、下属机构或外部合作方开放
- 需要内部核算、分摊、审计或月度锁账
- 希望把终端员工身份带入模型调用链

现阶段不主打

- 单一应用、单一团队的简单模型代理
- 面向公众的 API 充值、套餐与转售运营
- 追求最广泛的模型协议转换与模型服务覆盖
- 寻找完整的企业统一认证中心（IAM / SSO）
- 寻找 GPU Pod 放置、显存切分或模型副本级调度器

通用 API 数据面边界

该数据面默认关闭、按需开启；不打开时行为与 v1.9.x 完全一致，AI 链路零改动。AIGW 定位是身份、订阅、策略、审计与访问账的治理入口，不承担全站流量接入与七层负载的全部职责。

交付形态说明

AIGW 面向项目制的受控私有化交付：现场 POC、按客户环境部署、随版本提供升级与维护支持。

政企私有化　内容合规　API 治理入口　边界清晰

从下一次调用开始，把人、内容、规则与账说清楚

一个控制面治理 AI 调用和普通 API，每次调用有身份、有规则、有审核、有归属、有记录。

一次调用，六段都说得清

身份 ▸ 组织 ▸ 应用 / 模块 ▸ 策略 ▸ 模型 ▸ 用量账

谁在什么组织、通过什么应用与模块、基于什么规则、用了哪个模型、记在哪本账上。

管得住内容

关闭 / 观察 / 拦截三种模式，多阶段流水线，词库自主维护；命中只留脱敏摘要，查看动作也入审计。

追得到人

L1/L2 用于统计归因，L3 由服务端签名、网关验签，用于可信归账与追责。

算得清账

系统账、用户账、部门账，六维组合，快照冻结，月度锁账。

业务不中断

熔断回流、满载排队、Key 双活、模型灰度、节点排空；集群档演练在恒定 30 并发、非流式 POST 条件下实测 0 失败。

一套治理，两类流量

AI 调用与普通 API 共用身份、订阅、策略与审计；通用 API 数据面默认关闭、按需开启。

下一步：索取 POC 验收清单

清单列明验证范围、环境前提、逐项验收步骤与通过条件，可用于内部评估或直接安排现场 POC。来信即发。

重庆弘创达科技有限公司

官网：www.cqagc.com

邮箱：dengyao@cqagc.com

地址：重庆市两江新区栖霞路 18 号 13 幢 1 单元 11-7

AIGW　政企 AI + API 治理　内容合规　受控私有化交付