

API-Gateway API Gateway

Product Datasheet · Hardware model pending confirmation
ALN-DS-API-Gateway-EN · v1.1 · 2026-10-26 · English translation for review

Product overview

ALNETS API Gateway provides hardware for managing team AI access: cloud/local model sources, upstream accounts, proxies, employee model permissions, personal calling credentials and usage budgets. The product name is confirmed; the hardware model designation is not.

The application remains in development. Implementation, isolated tests, limited real calls and production delivery are distinct states; source-code capability or mock testing does not establish delivered system acceptance.

Core capabilities and application value

Capability	Function and value
Unified management	Manage models, sources, accounts, proxies, people and usage centrally.
Same-model account pools	Organize accounts and distribute requests by policy; production scheduling remains pending.
Account-specific proxies	Bind proxies per upstream account; verify real egress.
Model availability control	Check, publish, enable/disable and authorize models, distinguishing visibility from callability.
Personal credentials	Employees use gateway Keys rather than upstream credentials.
Usage and monthly budgets	Review Tokens by employee/model/date and configure admission/warning rules.

Typical business uses

Development teams, studios and enterprise projects requiring centralized model endpoints, credentials, access permissions and Token usage management.

01 / Models and Technical Specifications

Model configurations and interfaces

Item	Specification
Product name	ALNETS API Gateway
CPU	Intel Core i5; 4 cores; 3.8GHz
RAM	8GB DDR4
Storage	64GB SSD
Network	Dual Gigabit Ethernet
Motherboard	3.5-inch industrial-control motherboard
Supply	12V / 4A
Documented nominal power	25W; conditions pending
Dimensions (L × W × H)	118 × 113 × 39 mm
Package (L × W × H)	270 × 220 × 130 mm
Gross weight	1.1 kg
Runtime environment	Ubuntu 22.04 / 24.04 Server; Docker Compose

Technical notes

API Gateway manages service access; AiR supplies network ingress/egress and compute hosts supply local inference. Account egress depends on actual proxy resources, without bundled IPs or guarantees for all authentication paths or failure fallback.

Unconfirmed quantitative specifications

Hardware model, CPU model/generation, additional interfaces, electrical tests, environmental ratings and the formal software delivery matrix require confirmation.

02 / Sources, Account Pools and Proxies

Sources and model directory

Function	Behavior and current status
Source access	API Key / OAuth account management; implemented platform flows do not mean every platform passed real-call tests.
Local service registry	Add, edit or remove local service URLs and source models; removal does not stop the model host.
Directory publishing	Read models, import drafts, set scope, check/publish, enable/disable or delete.
Capability checks	Manage applicable interfaces/capabilities; source/proxy changes require rechecking earlier evidence.
Limited real routes	DeepSeek, Ternary and OpenAI OAuth have limited real-call records, specific to source/protocol.

Same-model multi-account pools

A fixed Sub2API revision was checked for multi-account scheduling. Current ALNETS OAuth v2 has pool binding/admin UI and reviewed isolated tests. Selection considers availability, priority, concurrency/load and session policy; equal round-robin is not guaranteed. Legacy v1 remains single-account; DeepSeek group paths allow multiple accounts.

Production enablement, real distribution, concurrency, failover and usage consistency remain pending. Where group calls cannot authoritatively identify the executing account, retain group-level attribution rather than inventing account-level usage.

Independent account proxies

HTTP, HTTPS, SOCKS5 and SOCKS5h configuration and manual tests are supported; API/OAuth accounts can explicitly bind/unbind proxies. Management/binding have test records; actual IP egress requires real-node verification.

Binding is per upstream account, not automatic public IP allocation per employee. It does not change the browser's network when opening OAuth authorization pages. IP resources are separate; all authentication paths and failure-fallback behavior are not fully accepted.

03 / Permissions, Usage and Version Status

Employees and credentials

Function	Configuration and behavior
Employee management	Administrators invite, enable/disable and grant selected or all models.
Personal calling Key	Issue, revoke and reissue; reissue revokes the previous valid Key.
Permission checks	Recheck employee/model access on calls; revocation affects subsequent requests, not necessarily already forwarded calls.
Employee directory	Show authorized models; external-client refresh timing depends on the client.

Token usage and budgets

Administrators filter complete Token records and paginated summaries by employee, model, date and real/mock source; employees see their own records. Employee, upstream-account and technical-Key ledgers are separate; overlapping entries must not be summed. Internal figures are not official provider invoices.

Budgets use Beijing calendar months, checking complete real usage before new calls and issuing warnings. With budgets enabled, an employee is limited to one unsettled real call. Unknown usage remains pending, not zero. A single request may overshoot; this is not an absolute currency or total-Token hard cap.

Development and delivery status

Management, permissions and some calls have slice-level test records. Native Codex/Claude Code protocols and Agent Host Runner have isolated candidates; real-provider/Host and production integration remain pending. Smart model selection is disabled by default. Real upgrade/rollback, complete billing, production HTTPS, backup/recovery and enterprise identity still have outstanding work.

The current management app uses React, Node.js and SQLite connected to Sub2API. Legacy PostgreSQL/Redis-only descriptions are not the complete software delivery list. Confirm formal versions, compatibility matrices and system concurrency per delivered release.

04 / Topology and Deployment Selection

Typical topology

```
Employee / app -> personal gateway Key -> authorized model
Cloud: account pool -> account -> bound proxy -> model service
Local: source binding -> LAN compute-host model endpoint
Admin -> accounts/proxies/models/permissions/usage/budgets
```

Scenarios and application value

Scenario	Configuration focus	Value
Development team	Central accounts and model directory	Unified employee calling identity.
Hybrid model project	Cloud + local source registry	Reduce per-client endpoint maintenance.
Multiple upstream accounts	Pools and proxy bindings	Organize resources and egress centrally.
Team usage	Employee budgets and Token queries	Establish reviewable usage rules.

Model and component selection

Confirm platform, authentication, model and client protocol before choosing a delivery version/hardware. Employee capacity, API throughput and SLA are not confirmed and cannot be inferred from CPU/RAM. AiR provides networking, compute hosts provide inference, and the gateway manages calls.

Deployment checks

Agree delivery scope/version -> deploy app and upstream -> configure sources/accounts/proxies -> verify real directory/calls -> set employee permissions/Keys -> reconcile Tokens/budgets -> accept pool, egress and failure paths.

Website: www.alnets.net **Email:** sale@alnets.net