External Provider API Gateway & Rate Governance

Doc type: technical · **Version:** v0.1 · **Status:** published · **Module slug:** external-provider-api-gateway-rate-governance
Exported: 2026-05-15 11:12 UTC · **By:** anonymous

External Provider API Gateway & Rate Governance – Technical Specification

1. Purpose of the Module

The External Provider API Gateway & Rate Governance module defines how Primoro safely, predictably, and fairly interacts with rate■limited external providers, independent of vendor.

Its purpose is to:

- enforce third■party API limits before violations occur
- protect patient and staff UX from upstream instability
- isolate interactive work from background and analytical loads
- ensure fairness across tenants, clinics, and workloads
- provide operational evidence during vendor escalation
- act as a deterministic control plane for all external integrations

This module is provider■agnostic by design and is a mandatory part of Primoro’s platform reliability layer.

2. Scope and Responsibilities

2.1 The External API Gateway & Rate Governance module is responsible for:

- Acting as the single ingress/egress control point for all rate■limited providers
- Enforcing:
- hard vendor ceilings
- tenant■level fairness
- workload class isolation
- Preventing burst amplification (pagination storms, sync loops, BI refreshes)
- Returning Primoro■owned, predictable error responses
- Emitting correlation IDs and quota telemetry
- Complementing backend:
- queues
- retries
- backoff
- circuit breakers

2.2 This module is not responsible for:

- Business logic or data transformation
- Vendor■specific authentication storage (Key Vault responsibility)
- Retry orchestration logic
- Job scheduling or batching
- Understanding provider semantics

This module governs when traffic is allowed, not what traffic does.

3. Core Architectural Principles

3.1 Control Plane vs Data Plane (Non■Negotiable)

- API Gateway (e.g. Azure API Management)
- rate limits
- fairness
- request shaping
- quota enforcement
- Provider Adapter Services
- retries
- exponential backoff
- circuit breaking
- idempotency
- deduplication

The gateway never connects directly to the provider API.

All external calls pass through Primoro adapter services .

3.2 Provider■Agnostic Design

No gateway policy may:

- reference provider■specific endpoints
- assume provider■specific limits
- rely on undocumented vendor behaviour

Provider limits are configured as data, not hardcoded.

3.3 Predictability Over Throughput

The system prefers:

- stable behaviour
- graceful degradation
- transparent throttling

over maximising throughput at the risk of upstream denial.

4. Traffic Classification (Mandatory)

All requests targeting external providers MUST be categorised into a traffic class.

4.1 Supported Traffic Classes

Every request must declare its class explicitly or be derived by route.

Unclassified traffic defaults to background.

5. Rate Governance Model

5.1 Composite Rate Keys

Rate limits are enforced using composite keys, ensuring isolation.

Canonical structure:

This enables:

- tenant isolation
- multi■provider support
- multi■token environments
- class■based fairness

5.2 Provider Hard Ceilings

Each provider defines a hard ceiling:

- per tenant
- per integration
- per time window

Hard ceilings are enforced before class budgeting.

5.3 Class Budgets (Fairness Layer)

Within provider ceilings:

- interactive traffic is protected
- background traffic is constrained
- BI traffic is tightly limited

BI traffic must never degrade interactive UX.

This model directly addresses real■world contention observed between live workflows and analytical refreshes .

5.4 Time■Window Guardrails

The gateway supports:

- per■minute limits

- per■hour quotas
- provider■specific windows

Typical use cases:

- “no more than N BI refreshes per hour”
- “background sync capped below interactive reserve”

5.5 Pagination & Burst Protection

Bulk and paginated flows must be explicitly flagged.

If flagged:

- tighter limits apply automatically
- burst amplification is dampened
- interactive capacity is preserved

6. Failure Handling & Degradation

6.1 Primoro■Owned Throttling Responses

When throttling occurs:

- HTTP 429 is returned
- response structure is Primoro■controlled
- correlation ID is always included

Clients can rely on consistent retry semantics, independent of provider behaviour .

6.2 Graceful Degradation

Rate governance:

- triggers before provider exhaustion
- prevents retry storms
- avoids cascading failures

Upstream failures never leak into unpredictable client behaviour.

7. Observability & Evidence

7.1 Mandatory Telemetry

The gateway logs:

- tenantId
- providerId
- traffic class
- quota key

- remaining allowance
- throttle events
- correlation IDs

This provides:

- live operations insight
- forensic traceability
- documented evidence for vendor negotiations .

8. Security & Isolation

- All calls are tenant■scoped
- No cross■tenant quota bleed
- No anonymous invocation
- Provider tokens are never exposed at gateway level
- Access Manager scoping is enforced indirectly via service identity

9. Integration Summary

This module underpins:

- PMS connector framework
- Payments providers
- Messaging providers
- Telephony providers
- Lab & imaging integrations
- Analytics pipelines
- Tenant health monitoring

It is part of Primoro’s platform resilience layer, not a customer■visible feature.

10. Explicit Non■Goals

This module does not:

- implement retries or backoff
- increase vendor quotas
- understand provider semantics
- schedule jobs
- prioritise traffic dynamically via AI

11. Versioning & Governance

This specification reflects:

- External Provider API Gateway & Rate Governance v2.0

All future changes must preserve:

- provider safety first
- interactive UX protection
- workload isolation
- predictability over throughput

12. Build Contract (Engineering & QA)

12.1 Canonical Rate Governance Model

Provider

- ProviderId
- DefaultCeiling
- SupportedWindows

RateKey

- TenantId
- ProviderId
- TrafficClass
- ProviderTokenId

TrafficClass

- interactive
- background
- bi

ThrottleEvent

- RateKey
- Limit
- Window
- Timestamp
- CorrelationId

12.2 Core Behaviour Rules

- All external calls pass through the gateway
- Provider ceilings are enforced first
- Traffic classes are enforced independently
- BI traffic cannot starve interactive traffic

- Pagination bursts are suppressed
- Throttle responses are predictable and owned by Primoro

12.3 Configuration Surfaces

Operators may configure:

- provider definitions
- per■provider ceilings
- class budgets
- time■window constraints
- pagination throttle behaviour
- route■to■class mappings

12.4 Acceptance Criteria

■ Provider rate limits are never exceeded

■ Interactive UX remains responsive

■ BI and background traffic are isolated

■ Throttle responses are deterministic

■ Correlation IDs support escalation

13. Non■Functional Requirements

Performance

- gateway adds negligible latency

Reliability

- failures fail fast, not slowly
- no cascading retry storms

Security & Governance

- strict tenant isolation
- immutable throttle logs
- safe behaviour under misclassification

End of Document

Embedded Tables

Table 1

Traffic Class	Description
interactive	End■user initiated flows (staff app, patient app, portal UX)
background	Scheduled syncs, imports, bulk processing
bi	Analytics, reporting, warehouse refreshes

Table 2

```
1
iderId}:{trafficClass}:{providerTokenId}
2
```