**4) Rate Limits & Tiering — “Fair, Burst-Friendly, and Honest”**

**Intent**  
Protect the platform while giving SMEs predictable throughput: burst + sustained budgets, clear backoff hints, and per-feature quotas.

**Actors**  
Rate Limiter • Tier Manager • Billing Meter • Tenants

**Preconditions**

* Tiers defined (Free/Standard/Gold/Enterprise) with budgets per endpoint.

**Flow**

1. **Budget Model**: per-tenant **sustained RPS** + **burst bucket** + **daily quota** (events/webhooks sized separately).
2. **Headers**: every response includes X-RateLimit-\* + **Backoff-Hint** (retry\_in\_ms, next\_window).
3. **Surge Protect**: if a tenant spikes beyond burst, auto-throttle gracefully; push **“you’re being throttled”** event.
4. **Grace Buckets**: temporary boost for onboarding or incident; expires automatically.
5. **Fairness Guard**: prevents noisy neighbor (cap % of global capacity per tenant).
6. **Metering & Alerts**: usage streamed to billing; alert tenants at 80/100% thresholds.

**Acceptance**

* Throttled clients see **deterministic** 429s with retry\_in\_ms.
* No tenant can starve others; fairness cap enforced.
* Webhooks never count against read budgets (separate pool).

**Innovations**

* **Backoff-Hint** headers (developer-friendly).
* **Grace Buckets** with auto-expiry.
* Separate **event budget** for reliability.