Step 7: Requests Observability — Logs, Metrics, and Live Tail

Goal: Add SOC■grade visibility into API usage: who connected, what endpoint they called, where they came from, and the outcome, with live tail and export.

New/Updated Endpoints

GET /v1/admin/requests	**Paginated, filtered view of request audit records (tenant-scoped).**
GET /v1/admin/requests/summary?window=15m	Counts for requests, 2xx/4xx/5xx, P95 latency, active clients.
GET /v1/admin/requests/stream	Server-Sent Events live tail of request records (filters supported).
GET /v1/metrics	Prometheus metrics include requests_total, request_duration_ms, active_clients

Response Schemas

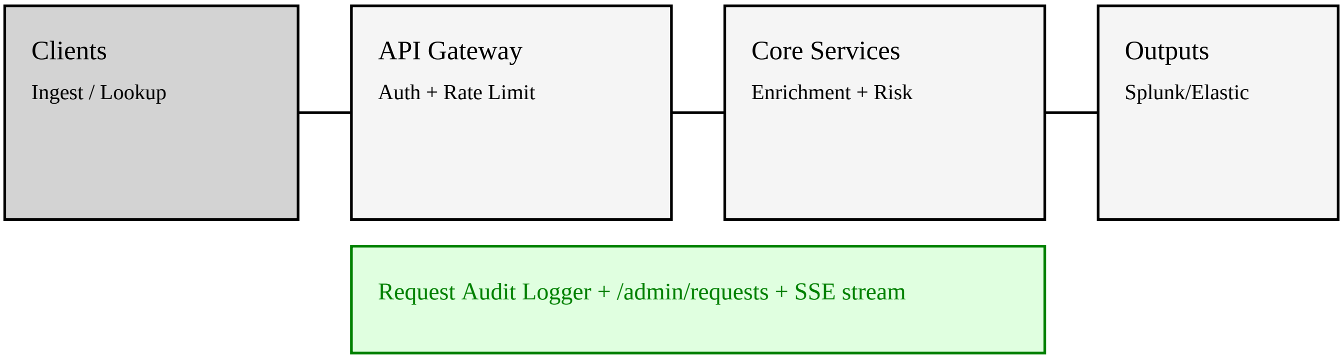
GET /v1/admin/requests — Example item

```
{
  "id": "b0b1...",
  "ts": "2025-08-14T12:45:11Z",
  "tenant_id": "tenant-123",
  "api_key_hash": "hmac:8c9e...",
  "client_ip": "203.0.113.9",
  "user_agent": "curl/8.6.0",
  "method": "POST",
  "path": "/v1/ingest",
  "status": 200,
  "duration_ms": 41,
  "bytes_in": 5241,
  "bytes_out": 238,
  "result": "ok",
  "trace_id": "c6f5a0...",
  "geo_country": "DE",
  "asn": "AS3320"
}
```

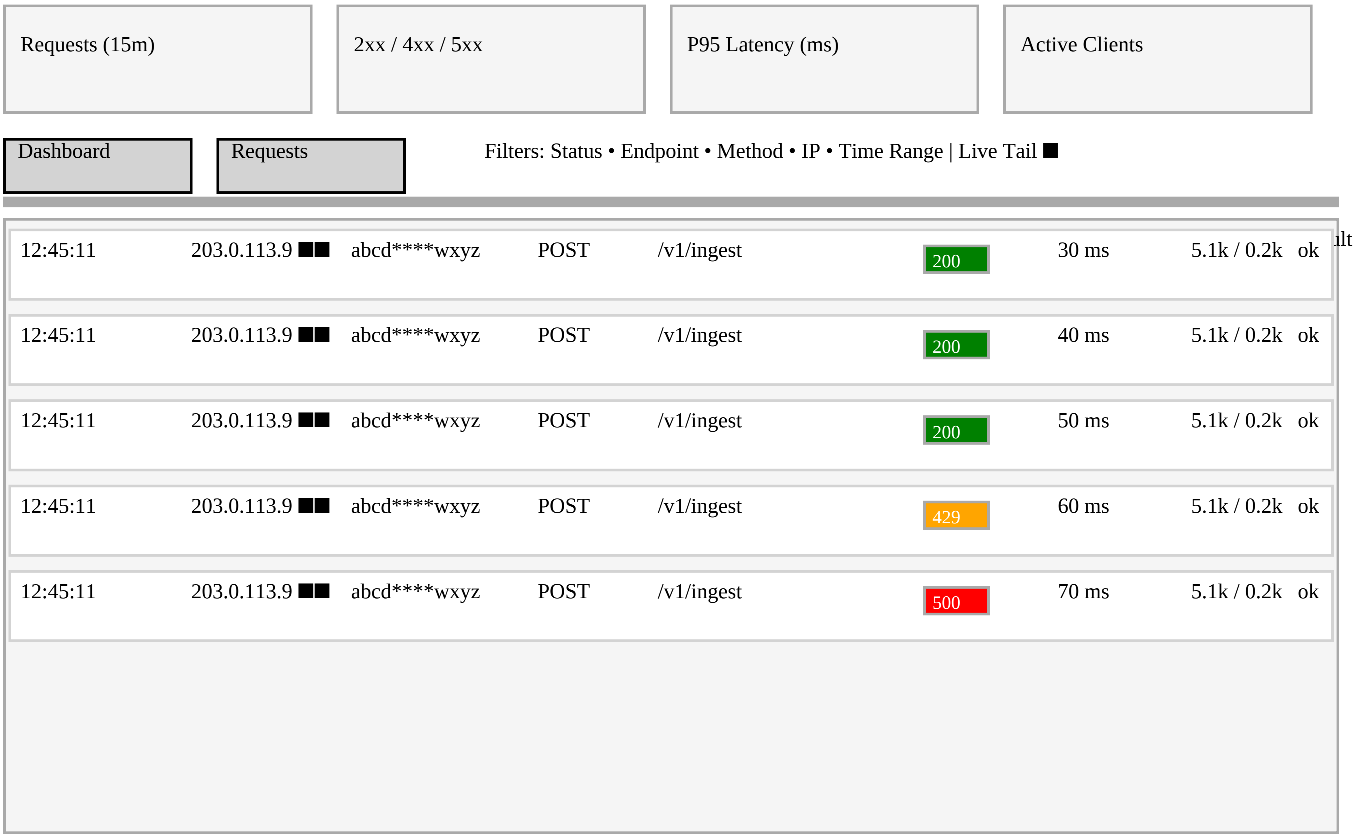
GET /v1/admin/requests/summary

```
{
  "requests": 1234,
  "codes": {"2xx": 1200, "4xx": 20, "5xx": 14},
  "p95_latency_ms": 42,
  "active_clients": 55
}
```

High-Level Architecture (Step 7 additions in green)



GUI Wireframes — Requests Observability



Requests (15m)

2xx / 4xx / 5xx

P95 Latency (ms)

Active Clients

Dashboard

Requests

Filters: Status • Endpoint • Method • IP • Time Range | Live Tail ■

ult

12:45:11	203.0.113.9 ■■	abcd****wxyz	POST	/v1/ingest	200	30 ms	5.1k / 0.2k	ok
12:45:11	203.0.113.9 ■■	abcd****wxyz	POST	/v1/ingest	200	40 ms	5.1k / 0.2k	ok
12:45:11	203.0.113.9 ■■	abcd****wxyz	POST	/v1/ingest	200	50 ms	5.1k / 0.2k	ok
12:45:11	203.0.113.9 ■■	abcd****wxyz	POST	/v1/ingest	429	60 ms	5.1k / 0.2k	ok
12:45:11	203.0.113.9 ■■	abcd****wxyz	POST	/v1/ingest	500	70 ms	5.1k / 0.2k	ok

Non-Functional Requirements

Performance	Audit logging must not add >1ms p50 overhead at 1k req/s; async writes; batch flush.
Security	Per-tenant scoping; HMAC hashing for API keys; never store request bodies.
Reliability	Backpressure on audit writes; drop policy for tail stream if slow consumer.
Retention	Automatic purge at 7 days to match MVP data policy.