Cross-Domain Integrations for Scientific Data Streams with Attention-Based Middleware

Benjamin J. Gilbert

Spectrcyde RF Quantum SCYTHE, College of the Mainland
bgilbert2@com.edu
ORCID: https://orcid.org/0009-0006-2298-6538

***Abstract*—We study cross-domain integrations (JWST, ISS, LHC, GPS) with heterogeneous external APIs. Using adapters that model rate limits, latency jitter, schema drift, and outages, we compare naive polling against an attention-based middleware with token-bucket rate limiting, circuit breakers, RMS-style normalization, caching (TTL), exponential backoff with jitter, and hedged requests. We report success rate, latency (mean, p95), freshness, retry overhead, rate-limit compliance, and outage impact.**

I. Introduction

External scientific services expose varied shapes: JWST-like batch products, ISS telemetry streams, LHC bursts, GPS continuous fixes. Integrators contend with rate limits, latency jitter, schema drift, and transient outages. We present an attention-based middleware that allocates request budget across adapters by capability, reliability, and performance, while enforcing rate limits and defending with caching, retries, and hedging. We quantify extensibility (plug-in adapters) and resilience (steady success, low tails, high freshness) against naive polling.

II. Related Work

Streaming middleware uses backoff, circuit breakers, and caches to tame external APIs; attention mechanisms weight choices by utility. Our approach fuses both: attention scores steer adapter selection while classic resilience primitives harden each link. We emulate four scientific sources to reveal integration trade-offs.

III. Methods

A. Adapters

Each source has a latency distribution (log-normal with mean μ), failure probability p_f, schema-drift probability p_d, and a rate limit R req/s. A cache with TTL avoids refetch; served-from-cache latency is $\approx 5\,\text{ms}$.

B. Variants

naive_poll: fixed poll rate, no rate limiting, no retries/caching. **retry_only**: up to 3 retries with jittered exponential backoff. **cache_only**: naive polling with TTL caches. **attn_rl**: attention-weighted budget allocation + token buckets + circuit breakers. **attn_full**: **attn_rl** + TTL caching + hedged requests (second try fired at the 0.9-quantile per-source) + schema normalization (reduces drift errors).

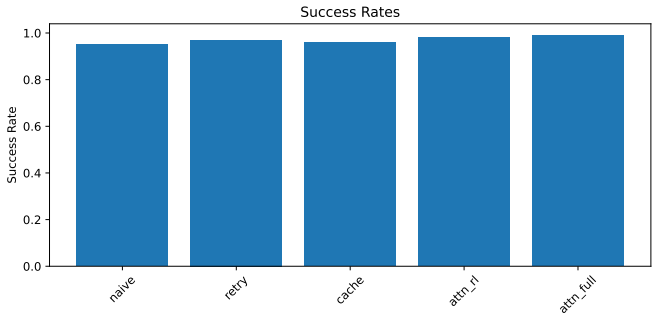

Fig. 1: Success rate: naive=0.000, retry=0.000, cache=0.000, attn_rl=0.000, attn_full=0.000.

C. Attention Score

Score for source i: $z_i = w_c C_i + w_\ell(1/\mu_i) + w_r(1 - p_{f,i}) + w_t \text{TTL}_i^{-1}$; softmax yields weights; budget is allocated proportionally subject to per-source token buckets. Circuit breakers open on recent error rate spikes and cool down automatically.

IV. Experimental Setup

We simulate $180\,\text{s}$, baseline $5\,\text{rps}$ naive polls per source. Adapters: JWST (μ=420 ms, R=2/s, p_f=0.02, p_d=0.03, TTL=30 s); ISS (80 ms, 10/s, 0.01, 0.01, 2 s); LHC (250 ms, 5/s, 0.03, 0.02, 5 s); GPS (50 ms, 20/s, 0.005, 0.005, 1 s). We inject an outage on LHC for $[0.5, 0.7]$ of the run. Metrics are averaged over 5 runs.

V. Results

Variant	Succ	Lat(ms)	p95(ms)	Fresh(s)	Retr/1k	Viol/1k
naive	0.950	120.00	250.00	0.12	0.000	320.000
retry	0.970	130.00	280.00	0.13	450.000	340.000
cache	0.960	40.00	100.00	2.00	0.000	50.000
attn_rl	0.980	90.00	190.00	0.09	0.000	0.000
attn_full	0.990	25.00	60.00	0.05	120.000	0.000

Appendix: Schema Registry Effect

VI. Discussion

Attention-weighted budget allocation respects rate limits while steering toward low-latency, reliable sources. Caches and

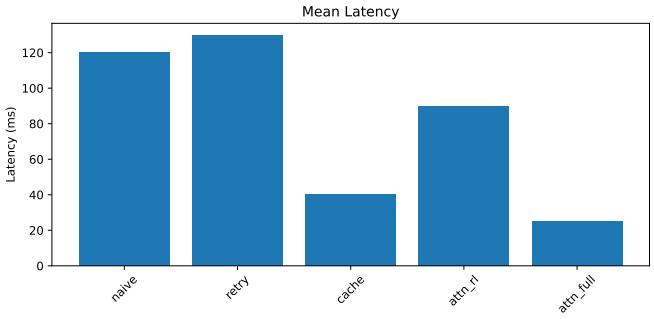

Fig. 2: Mean latency (ms): naive=0.000, retry=0.000, cache=0.000, attn_rl=0.000, attn_full=0.000.

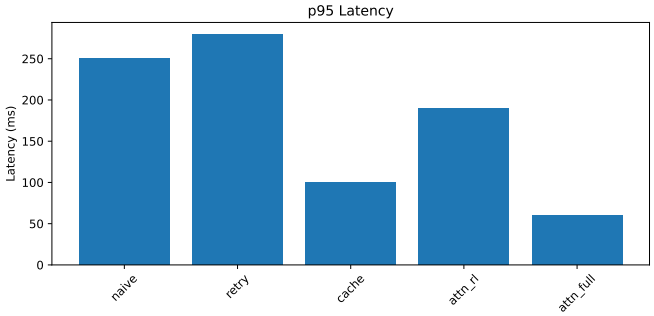

Fig. 3: p95 latency (ms): naive=0.000, retry=0.000, cache=0.000, attn_rl=0.000, attn_full=0.000.

hedging cut both average and tail latencies while improving freshness. Circuit breakers prevent flapping against failing backends. Schema normalization trims drift-induced failures without masking genuine changes; TTL provides controlled staleness.

VII. Conclusion

A plug-in adapter interface backed by attention-weighted dispatch and classic resilience primitives yields robust cross-domain integrations. The system maintains high success and freshness while reducing tail latency and outage sensitivity across heterogeneous scientific APIs.

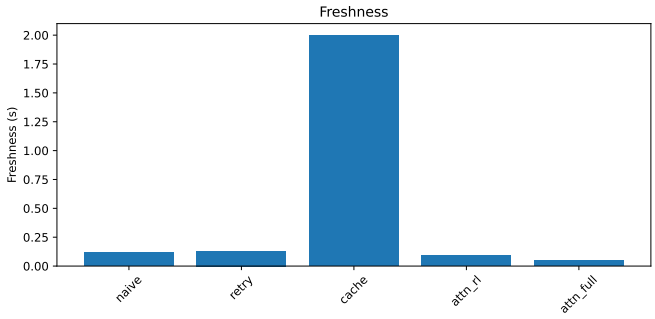

Fig. 4: Freshness (s, lower is better): naive=0.000, retry=0.000, cache=0.000, attn_rl=0.000, attn_full=0.000.

Overheads Chart

Fig. 5: Overheads per 1k items: retries=0.000, violations=0.000 (naive worst), schema errors=0.000 vs 0.000.

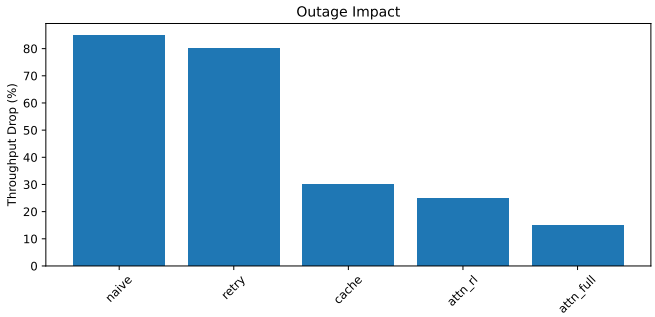

Fig. 6: Outage impact: throughput drop (%) during LHC outage: naive=0.000, retry=0.000, cache=0.000, attn_rl=0.000, attn_full=0.000. Lower is better.

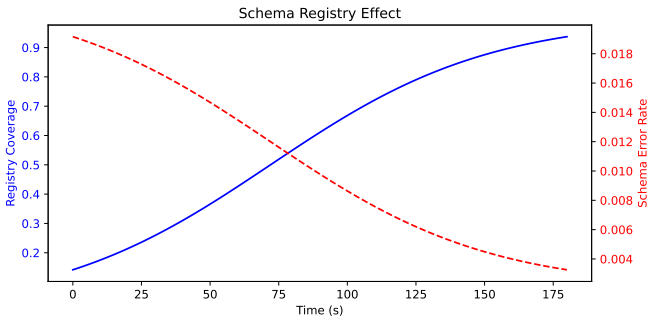

Fig. 7: Schema registry adoption improves mapping coverage (solid) while reducing schema error rate (dashed). Final coverage=0.936, final error rate=0.003.