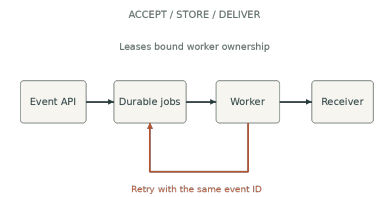


BACKEND SERVICES

Webhook delivery queue

Keep a temporary endpoint failure from losing a business event.



PROPOSED ARCHITECTURE

Delivery design; retry policy to specify

THE PROBLEM

An application emits events faster than a downstream service can accept them. Sending directly from the request handler couples both services. A delivery queue stores the event first, then tracks each attempt and gives an operator a controlled replay path.

SCOPE

- Store incoming events and delivery jobs in PostgreSQL; workers claim expiring leases.
- Use a TypeScript API and worker with signed requests, bounded backoff, and per-endpoint limits.
- Provide an attempt log, paused endpoints, and replay of exhausted jobs.

PLANNED DELIVERABLES

- API, worker, schema, and local receiver simulator
- Crash, timeout, signature, and concurrency tests
- Replay UI and an operations guide with failure procedures

CONSTRAINTS

Delivery

At least once; receivers must deduplicate stable event IDs

Destinations

Administrator-approved endpoints only

Recovery

Expired leases return interrupted jobs to the queue

Retention

Bounded payload retention; credentials excluded from logs

VERIFICATION PLAN

Lost response

Accept the event, drop the response, and verify the retry keeps its ID.

Worker crash

Kill a worker after claim; another worker recovers the expired lease.

Backpressure

A slow receiver must not consume every worker or block other endpoints.

NEXT STEP

Define the event envelope, endpoint trust model, and retry budget.

Delivery is at least once. Retention limits how long events can be recovered; receivers must handle duplicate IDs.