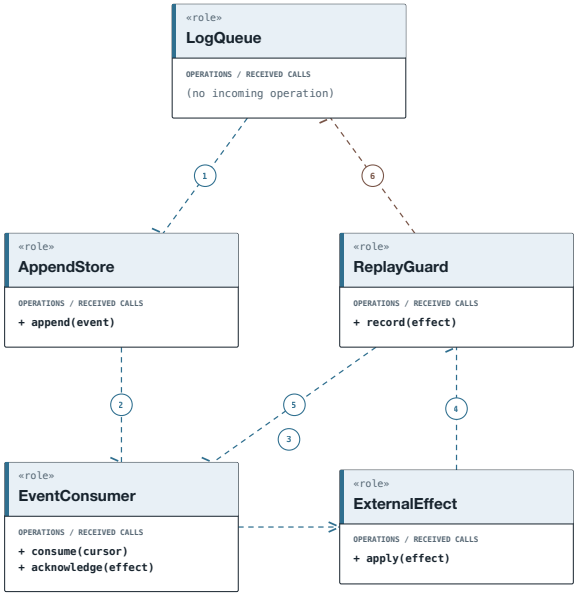
P13 Log-as-queue loops

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Drive work from a durable execution record.

UML CLASS VIEW

dashed arrow: call dependency · brown: return/failure · analytical, not an observed trace



POINTCUT / PATTERN INTERFACE

CONTEXT	New work is represented by durable append-only records.
TRIGGER	A consumer polls or subscribes to unseen records.
STRUCTURE	Event identity, append order, consumer cursor, and acknowledgement.
EFFECT	Work advances from the log without replaying already completed effects.

CONTRACT / REQUIRED BEHAVIOR

REQUIRES	Records have stable identity and consumers retain progress.
ENSURES	Each effectful unit has an idempotency or acknowledgement rule.
INVARIANT	Consumer crash does not erase unacknowledged work.
VIOLATION	Duplicate delivery is deduplicated or documented as at-least-once.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 LogQueue → AppendStore append(event)
2 AppendStore → EventConsumer consume(cursor)
3 EventConsumer → ExternalEffect apply(effect)
4 ExternalEffect → ReplayGuard record(effect)
5 ReplayGuard → EventConsumer acknowledge(effect)
6 ReplayGuard ↩ LogQueue replay(cursor)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	SQL log, file log, queue adapter, polling, and projection.
TEST_OBLIGATION	Test crash before ack, duplicate delivery, out-of-order reads, and replay.
NEGATIVE_CONTROL	An in-memory queue is not log-as-queue.