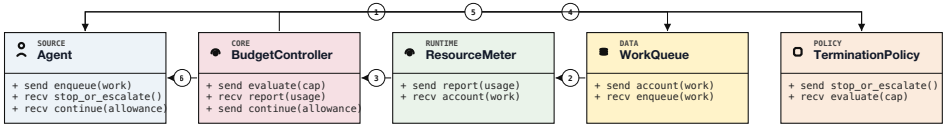
P19 Budgeted loop with caps

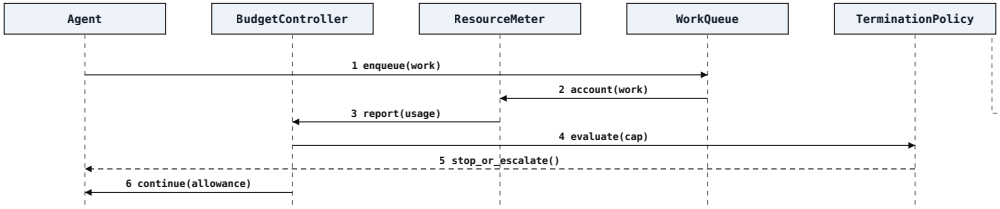
Primitive · A² A2 · gauge role collaboration

Bound execution resources.

STRUCTURE · PARTICIPANT COLLABORATION



INTERACTION



MESSAGE TRACE 1 enqueue(work) → 2 account(work) → 3 report(usage) → 4 evaluate(cap) → 5 stop_or_escalate() → 6 continue(allowance)

PATTERN INTERFACE

context
The harness can consume bounded turns, steps, tokens, time, or cost.

trigger
Accounting approaches or exceeds a configured cap.

structure
Resource ledger, cap policy, terminal reason, and optional escalation.

effect
Execution stops or changes policy before an unaccounted resource overrun.

CONTRACT ANNOTATIONS

requires	Relevant resource use is observable and shared with delegated work.
ensures	Exhaustion produces a distinct terminal outcome and records the responsible cap.
invariant	Budget exhaustion is never reported as task success.
violation	Missing accounting or handler failure falls back to a safe terminal policy.

Evidence: 30/41 repositories, 27/36 families; 30 trajectories (17 multi-checkpoint); exact audit: not sampled.

Variation: Step, turn, token, cost, wall-clock, delegation, and per-task budgets.

Negative control: A constant limit in an unrelated library does not meet the pointcut.