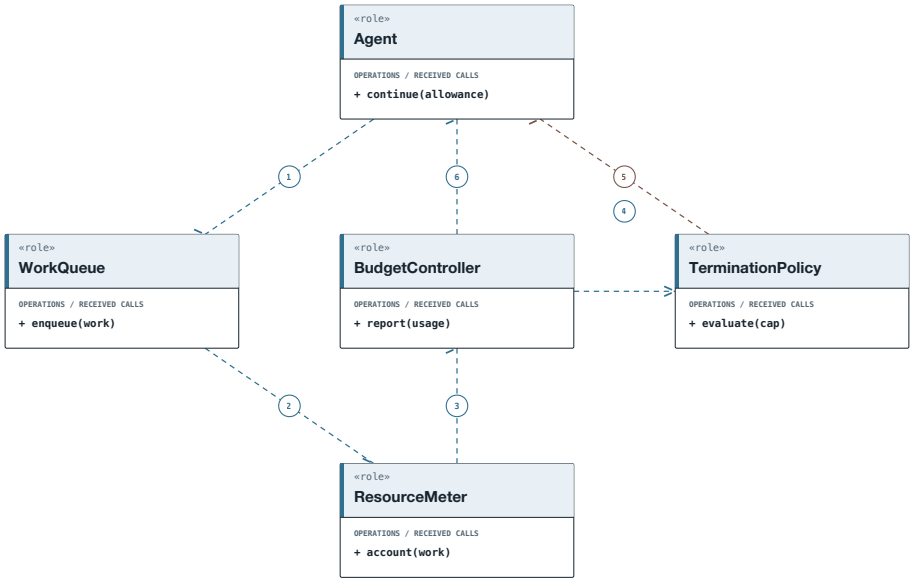
P19 Budgeted loop with caps

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Bound execution resources.

UML CLASS VIEW

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



POINTCUT / 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 / REQUIRED BEHAVIOR

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.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 Agent → WorkQueue enqueue(work)
2 WorkQueue → ResourceMeter account(work)
3 ResourceMeter → BudgetController report(usage)
4 BudgetController → TerminationPolicy evaluate(cap)
5 TerminationPolicy ↩ Agent stop_or_escalate()
6 BudgetController → Agent continue(allowance)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	Step, turn, token, cost, wall-clock, delegation, and per-task budgets.
TEST_OBLIGATION	Test exact cap, overshoot, parallel accounting, child work, and custom handler.
NEGATIVE_CONTROL	A constant limit in an unrelated library does not meet the pointcut.