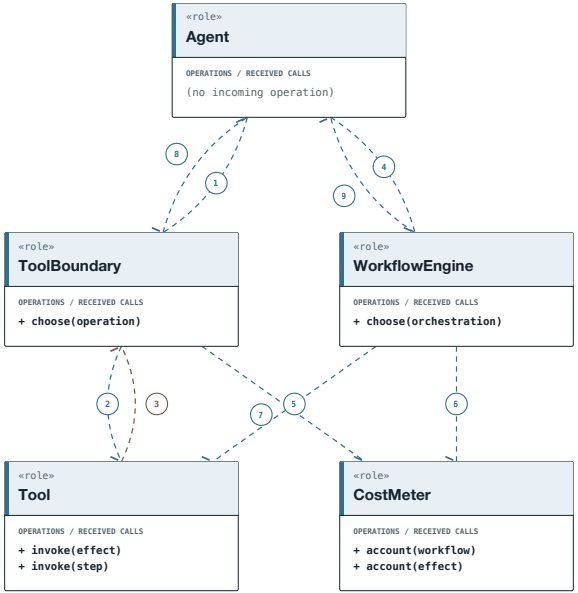
P11 Tool-workflow boundary

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Distinguish an operation from an orchestration.

UML CLASS VIEW

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



POINTCUT / PATTERN INTERFACE

CONTEXT	The harness exposes both primitive effects and multi-step procedures.
TRIGGER	The agent chooses a tool or workflow entry point.
STRUCTURE	Separate effect/result contracts and workflow ownership of sequencing.
EFFECT	Workflow orchestration is not mistaken for a single external side effect.

CONTRACT / REQUIRED BEHAVIOR

REQUIRES	Tool and workflow descriptors have distinct schemas and execution paths.
ENSURES	Retry, budget, and result accounting bind to the correct unit.
INVARIANT	A workflow cannot hide duplicate irreversible tool effects.
VIOLATION	Nested failures and retries produce explicit ownership and cost records.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 Agent → ToolBoundary choose(operation)
2 ToolBoundary → Tool invoke(effect)
3 Tool ↩ ToolBoundary return(effect)
4 Agent → WorkflowEngine choose(orchestration)
5 WorkflowEngine → Tool invoke(step)
6 WorkflowEngine → CostMeter account(workflow)
7 ToolBoundary → CostMeter account(effect)
8 ToolBoundary → Agent return(result)
9 WorkflowEngine → Agent return(result)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	Workflow DSL, composite tool, procedure service, and sub-agent workflow.
TEST_OBLIGATION	Test retry inside workflow, duplicate steps, and result aggregation.
NEGATIVE_CONTROL	A single function that happens to call another function is not enough.