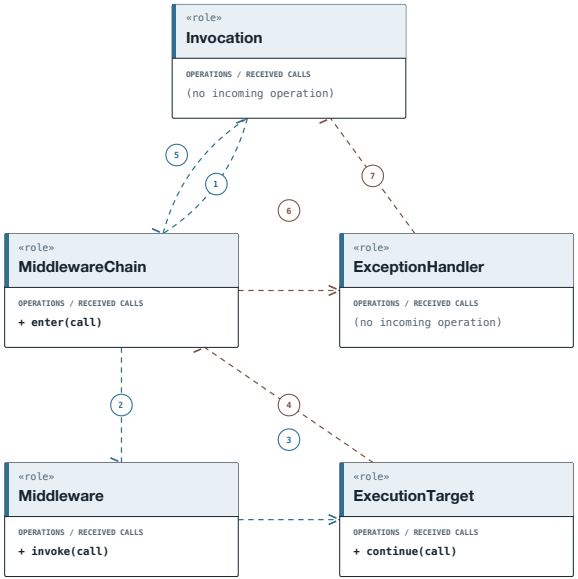
P04 Middleware-pipeline loop

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Layer behavior around model/tool calls.

UML CLASS VIEW

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



POINTCUT / PATTERN INTERFACE

CONTEXT	A harness surrounds model or tool invocation with extensible behavior.
TRIGGER	A call enters or leaves the runtime boundary.
STRUCTURE	Ordered middleware entries with a continuation or short-circuit result.
EFFECT	Logging, policy, retry, or context behavior runs in declared order.

CONTRACT / REQUIRED BEHAVIOR

REQUIRES	The boundary exposes a composable continuation contract.
ENSURES	Middleware executes in deterministic order and the final result reaches the loop.
INVARIANT	A middleware does not silently bypass required downstream layers.
VIOLATION	Short circuits and exceptions unwind deterministically with one terminal result.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 Invocation → MiddlewareChain enter(call)
2 MiddlewareChain → Middleware invoke(call)
3 Middleware → ExecutionTarget continue(call)
4 ExecutionTarget ↩ MiddlewareChain return(result)
5 MiddlewareChain → Invocation return(result)
6 MiddlewareChain ↩ ExceptionHandler unwind(error)
7 ExceptionHandler ↩ Invocation fail(error)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	Registration order, retry, redaction, tracing, and policy middleware.
TEST_OBLIGATION	Test order, short circuit, exception, and repeated side-effect calls.
NEGATIVE_CONTROL	A hard-coded helper call is not a middleware pipeline.