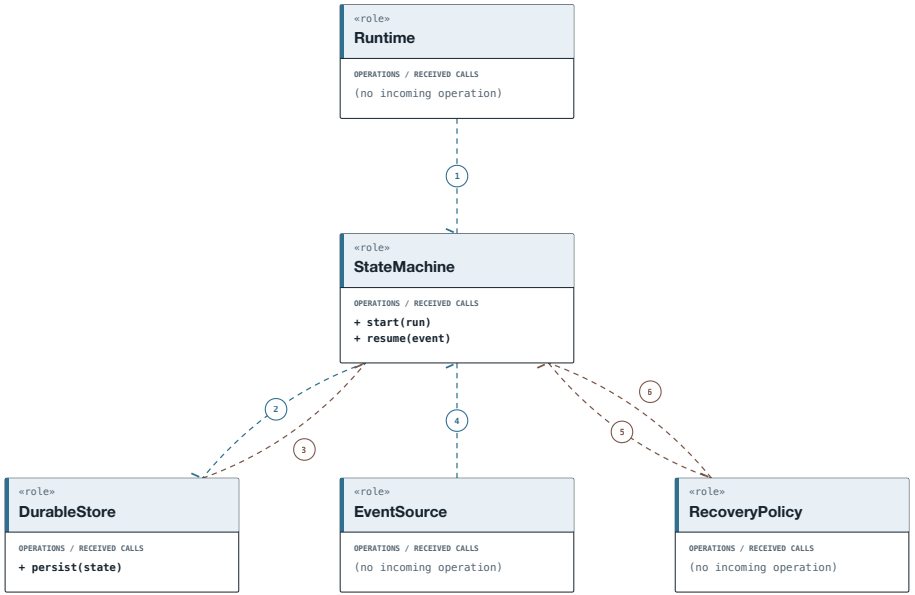
P03 Asynchronous state machine

Primitive / UML CLASS VIEW / ANALYTICAL MODEL

Make waiting and resumption explicit.

UML CLASS VIEW

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



POINTCUT / PATTERN INTERFACE

CONTEXT	A harness run may pause for external work or user input.
TRIGGER	An event requests suspension or arrives to resume a paused run.
STRUCTURE	Enumerated states, legal transitions, event identity, and resumption state.
EFFECT	A resumed run follows only the legal transition and preserves durable state.

CONTRACT / REQUIRED BEHAVIOR

REQUIRES	State can be serialized or retained and incoming events can be identified.
ENSURES	Accepted events move the machine to a defined state and duplicate events are handled.
INVARIANT	No transition crosses from a disallowed state or executes twice after replay.
VIOLATION	Invalid, duplicate, or interrupted events leave a recoverable state or raise explicitly.

RELATIONSHIPS / ANALYTICAL CALLS

number · caller → callee · message

1 Runtime → StateMachine start(run)
2 StateMachine → DurableStore persist(state)
3 DurableStore ↩ StateMachine restore(state)
4 EventSource → StateMachine resume(event)
5 StateMachine ↩ RecoveryPolicy on_error(error)
6 RecoveryPolicy ↩ StateMachine recover(state)

VARIATION / TEST / NEGATIVE CONTROL

VARIATION	Transition dispatch, persistence backend, and recovery policy.
TEST_OBLIGATION	Test illegal transitions, duplicate events, interruption, and resume.
NEGATIVE_CONTROL	An enum without transition enforcement is not sufficient.