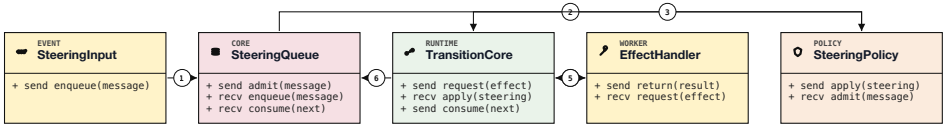
P06 Functional core and steering queues

Primitive · A² A1 · queue role collaboration

Separate transitions from live control.

STRUCTURE · PARTICIPANT COLLABORATION



PATTERN INTERFACE

context
A run accepts steering or queued input while transitions continue.

trigger
A new steering message or input item is enqueued.

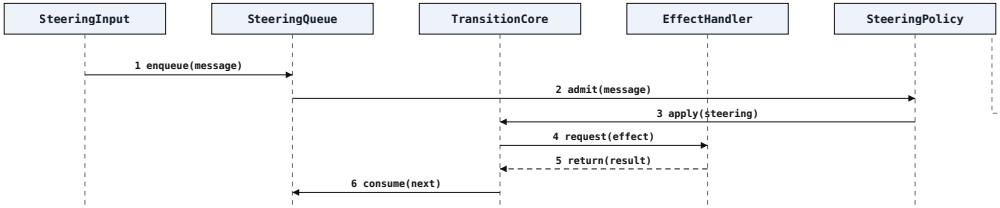
structure
A transition core plus queued control messages and defined application points.

effect
The transition consumes or defers steering with explicit ordering semantics.

CONTRACT ANNOTATIONS

requires	Steering messages have identity and a defined queue boundary.
ensures	Transition state changes only through the core and replayable control inputs.
invariant	Live input cannot mutate shared state nondeterministically outside the queue.
violation	Queue overflow, conflict, or replay produces explicit policy outcomes.

INTERACTION



MESSAGE TRACE 1 enqueue(message) → 2 admit(message) → 3 apply(steering) → 4 request(effect) → 5 return(result) → 6 consume(next)

Evidence: 26/41 repositories, 25/36 families; 26 trajectories (13 multi-checkpoint); exact audit: not sampled.

Variation: In-memory or durable inbox, priority, coalescing, and effect handlers.

Negative control: A mutable global message list without transition semantics is insufficient.