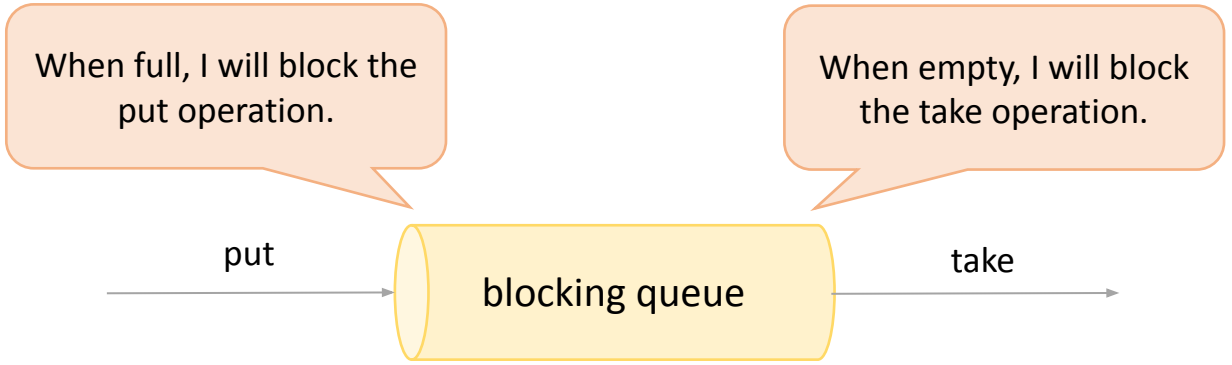
Blocking queue and producer-consumer pattern



helps implement patterns like

backpressure

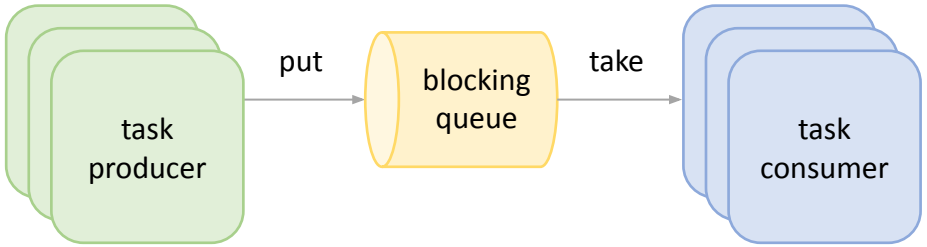
notify clients and ask them (or even force) to slow down data publishing rate

long polling

broker blocks pull request from consumer and waits for messages to arrive

producer-consumer

separate production of tasks from their execution



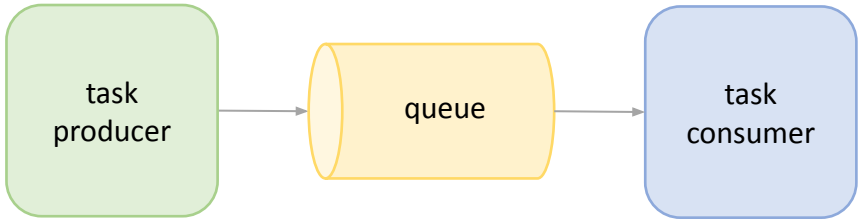
Blocking queue and producer-consumer pattern

without blocking queues
we can implement the producer-consumer pattern
by using

pauses execution
of a thread

wakes up
the waiting thread

wait and notify



call wait when full

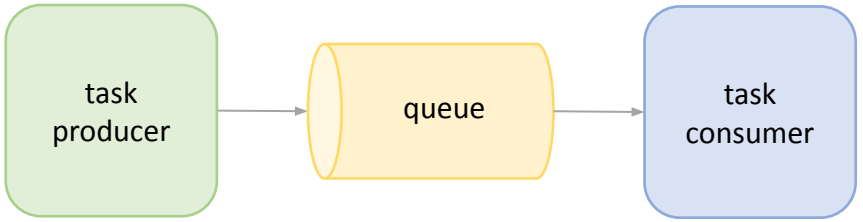
call notify

call notify

call wait when empty

variable that holds a set of permits,
a counter

semaphores



occupied spots counter

empty spots counter

`emptyCount--`
`fillCount++`

thread is blocked when
`emptyCount` becomes 0

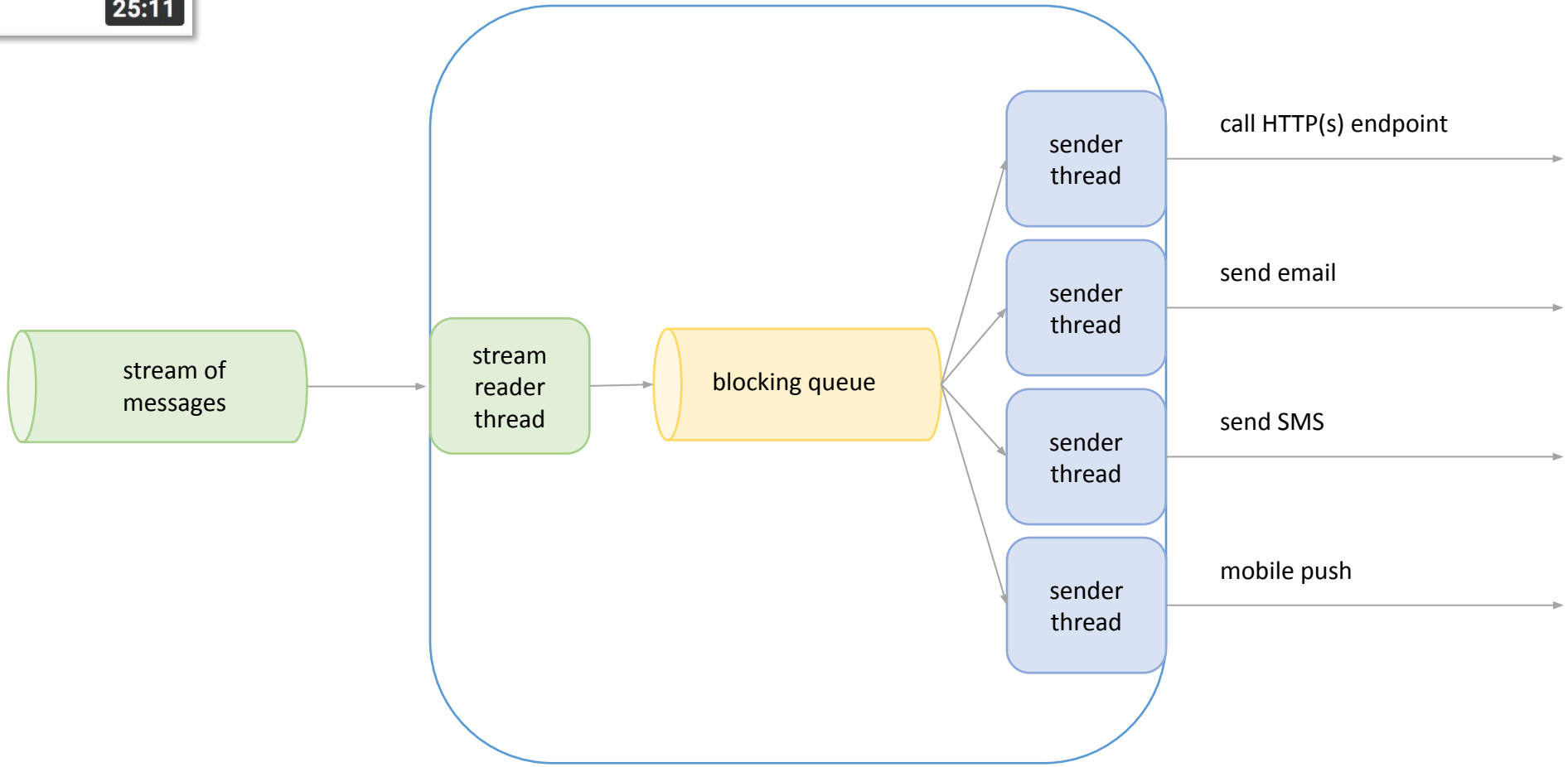
`fillCount--`
`emptyCount++`

thread is blocked when
`fillCount` becomes 0

Blocking queue and producer-consumer pattern

blocking queue applications

notification system



Blocking queue and producer-consumer pattern

blocking queue applications

data aggregation

(e.g. during stream processing)