

Concurrent Programming Using The Disruptor

Trisha Gee, Developer at LMAX

@trisha_gee

mechanitis.blogspot.com

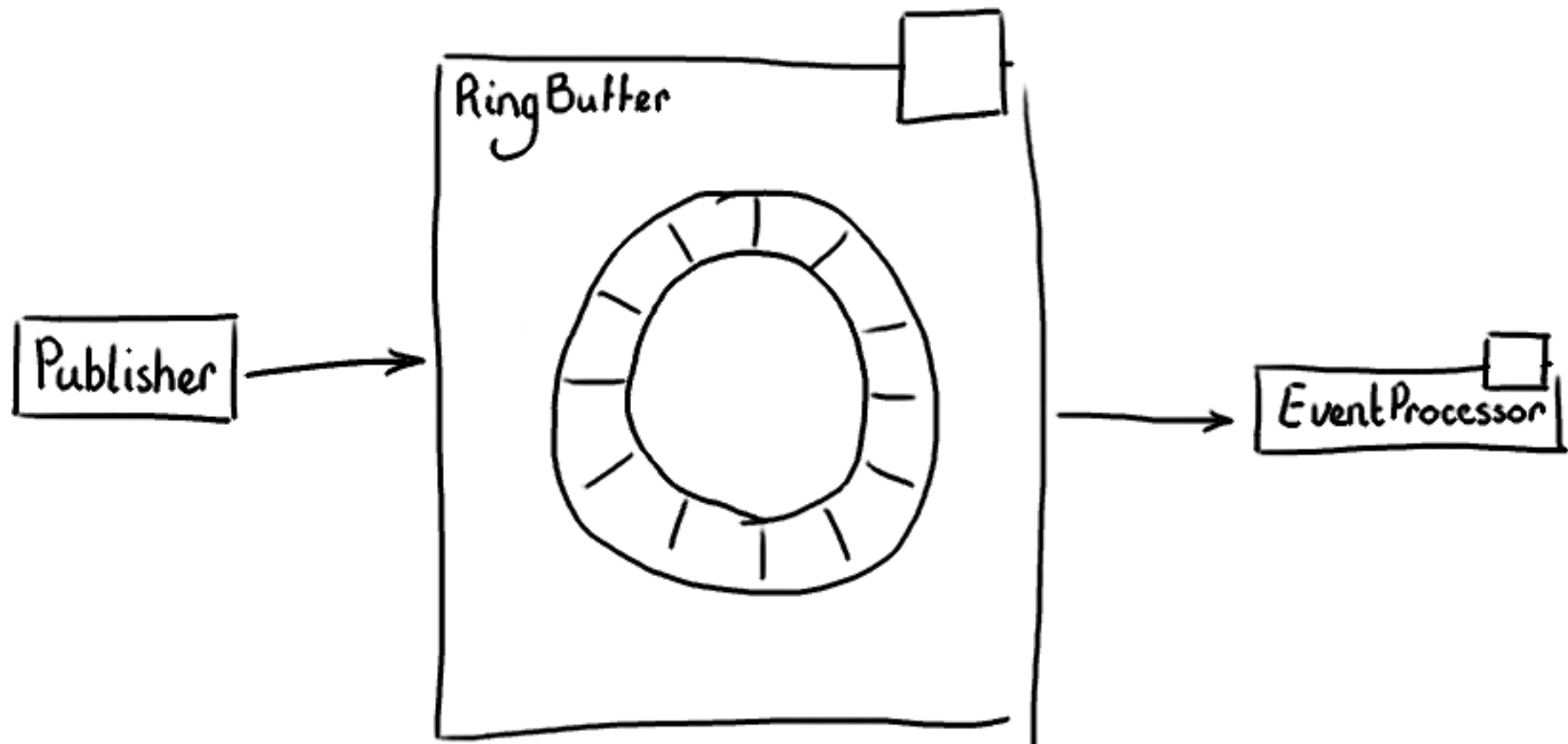


The Disruptor?

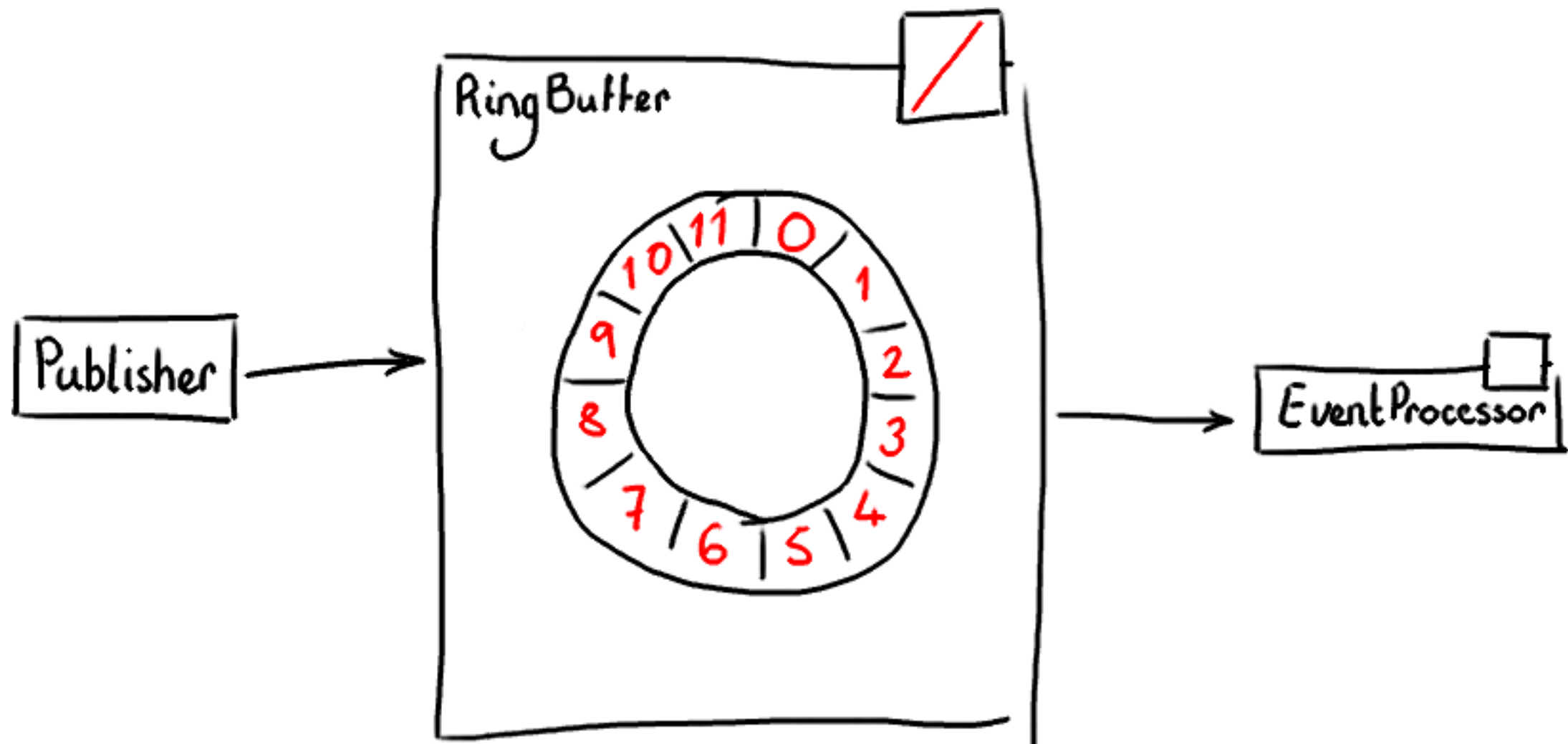
What is it?

- Data structure and work flow with no contention.
- Very fast message passing.
- Allows you to go truly parallel.

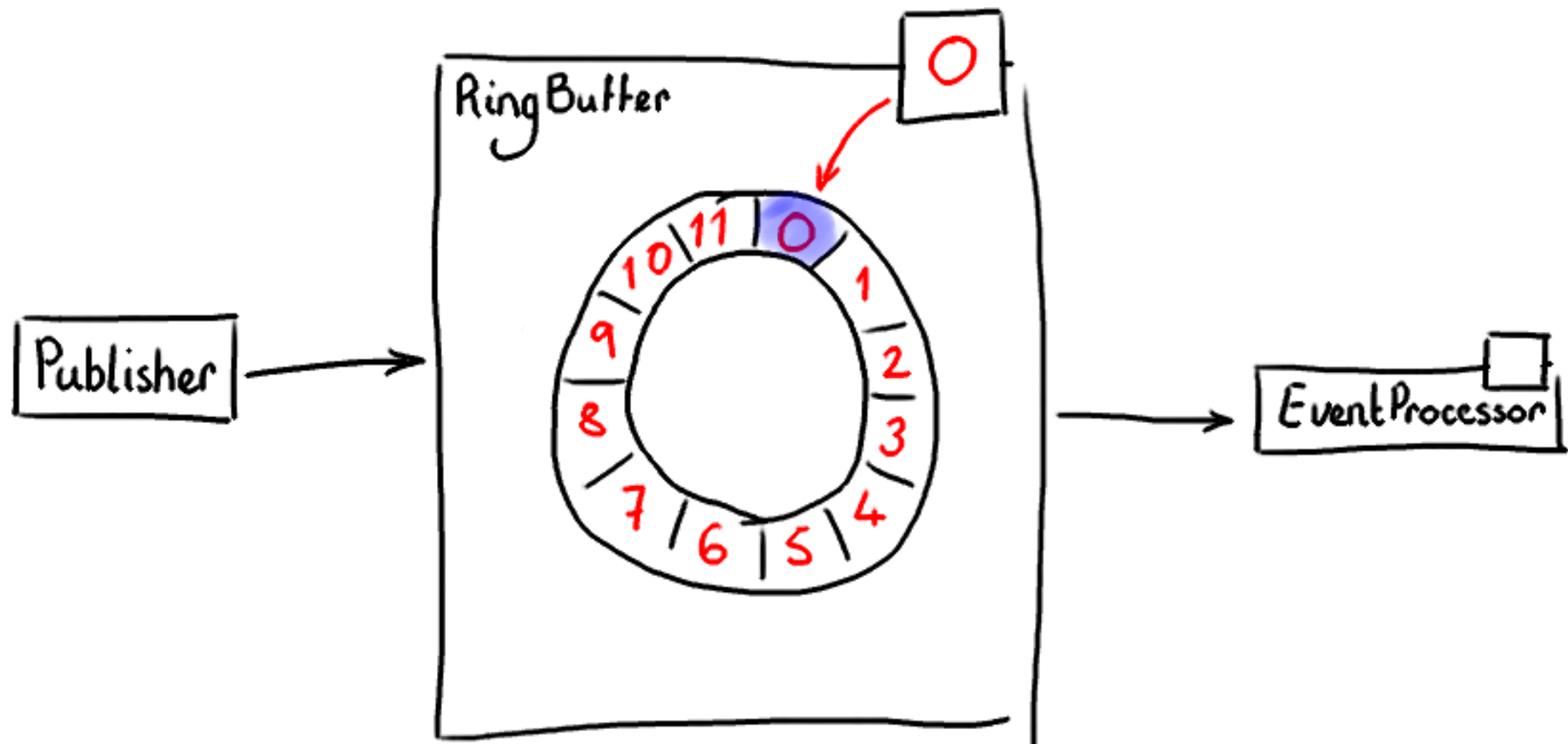
The Magic RingBuffer



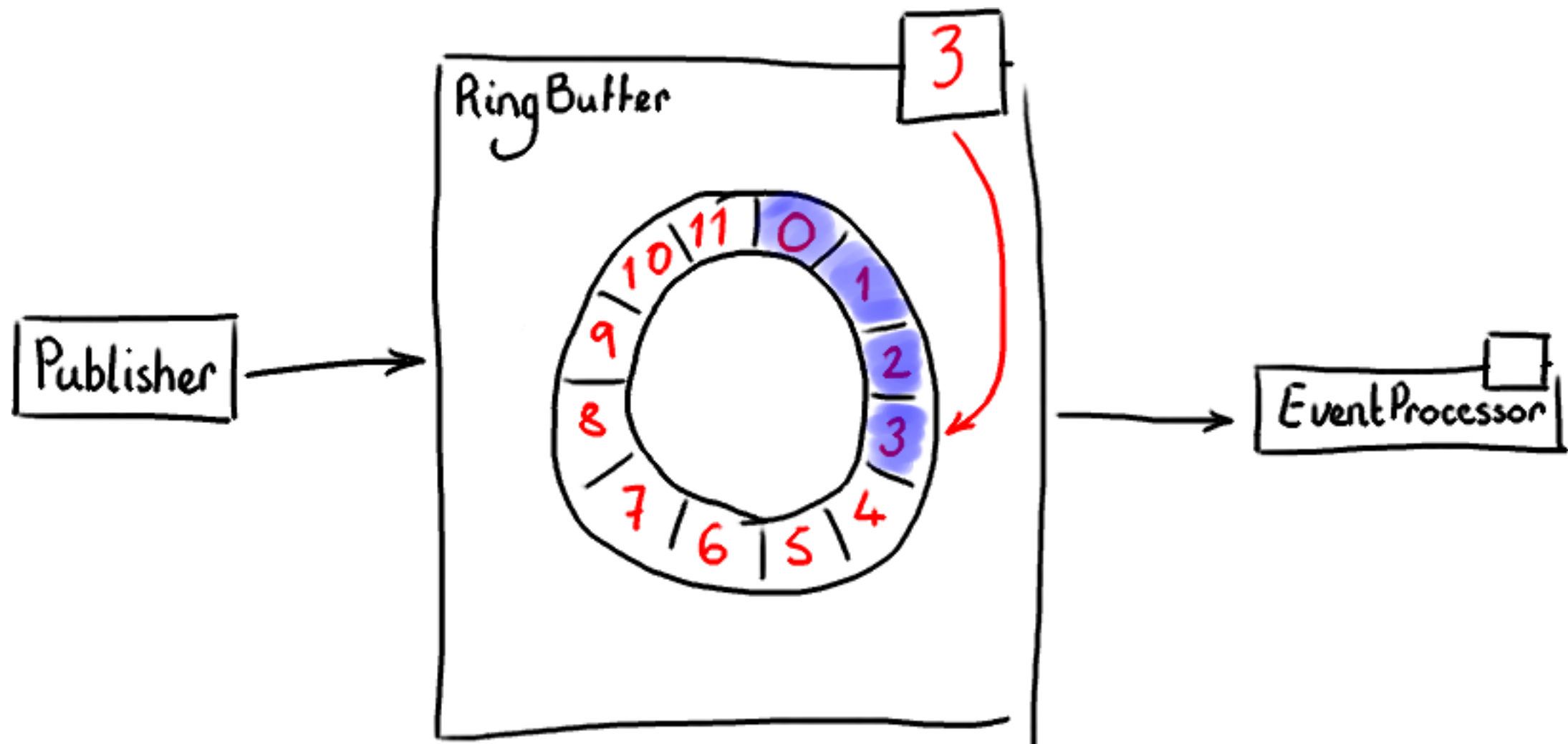
The Magic RingBuffer



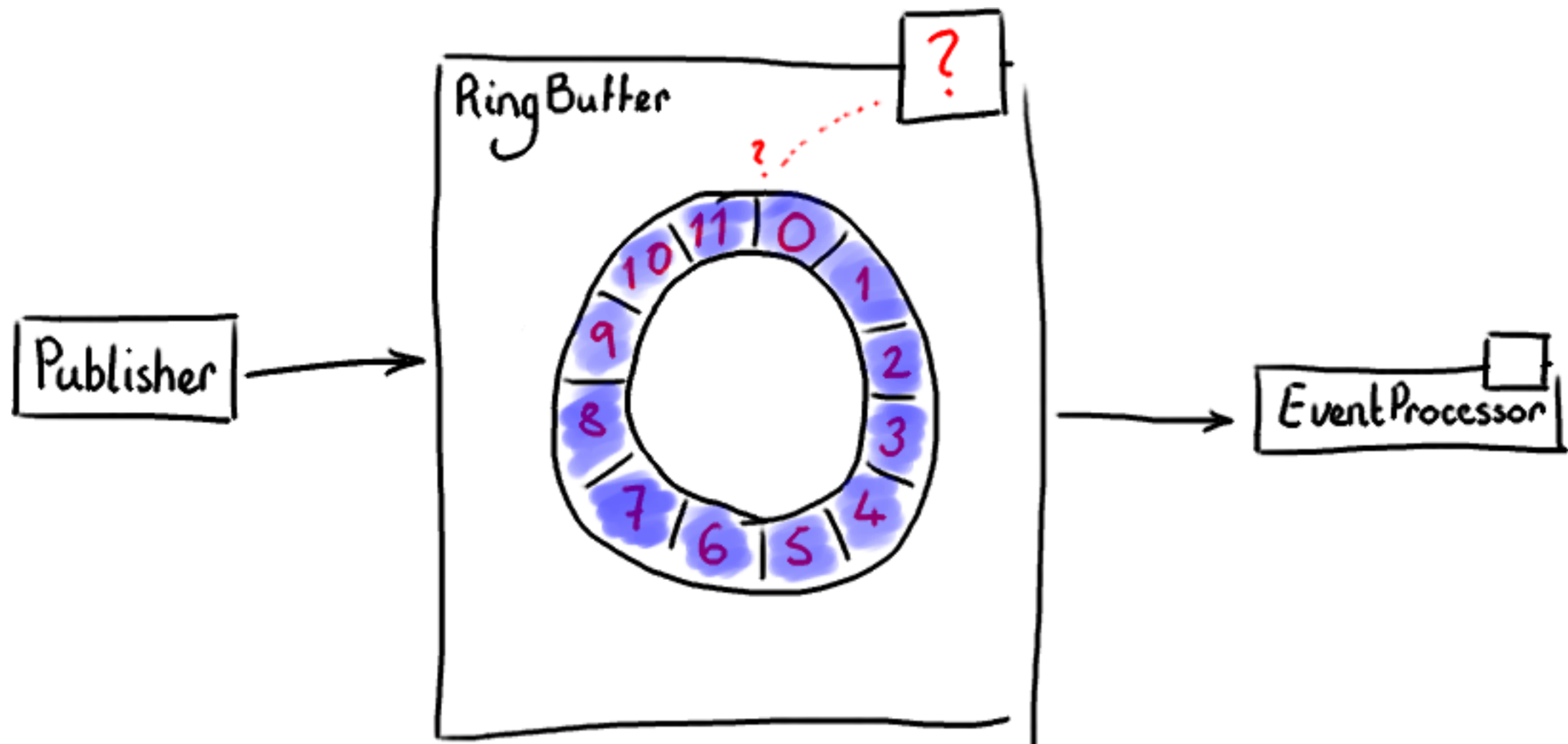
The Magic RingBuffer



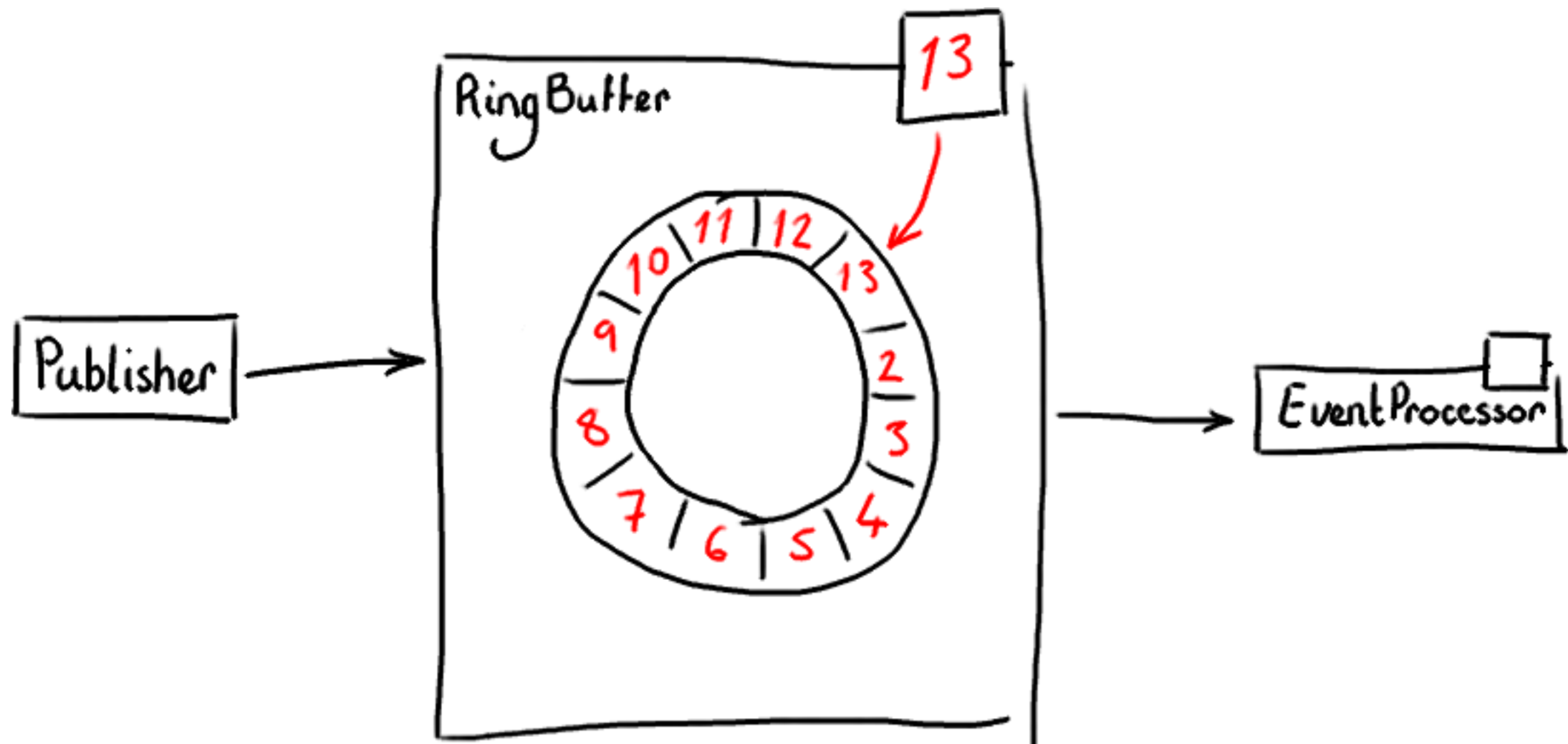
The Magic RingBuffer



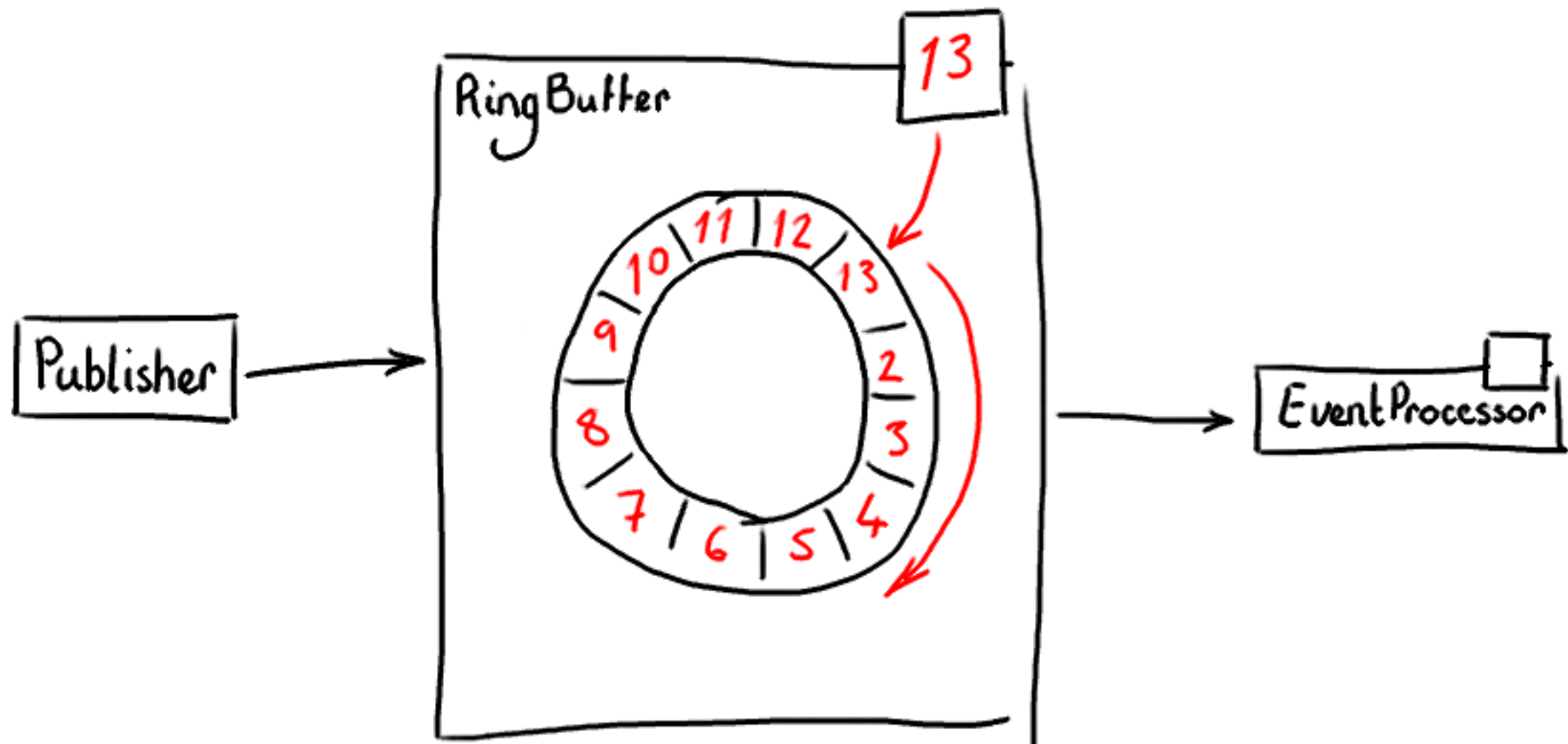
The Magic RingBuffer



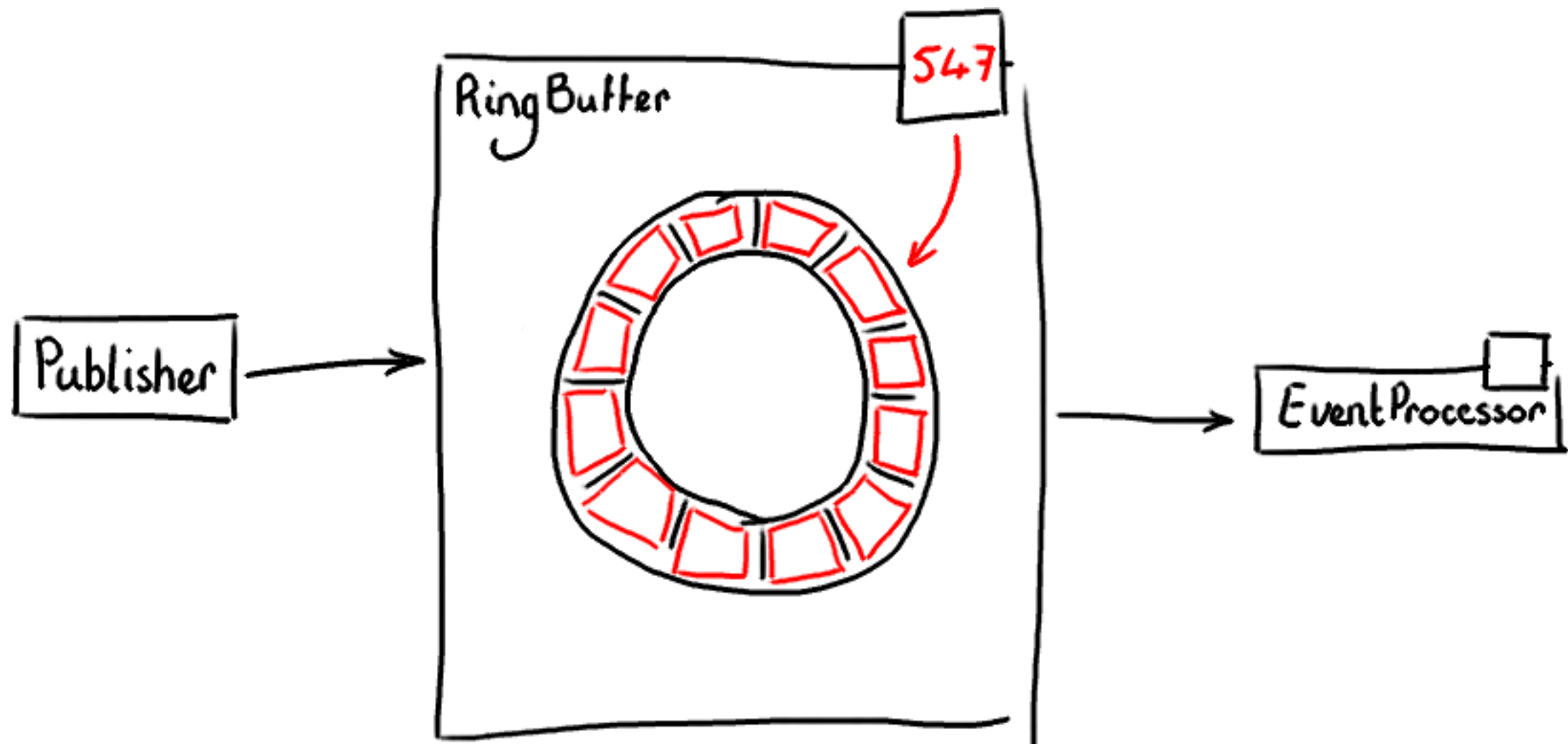
The Magic RingBuffer



The Magic RingBuffer



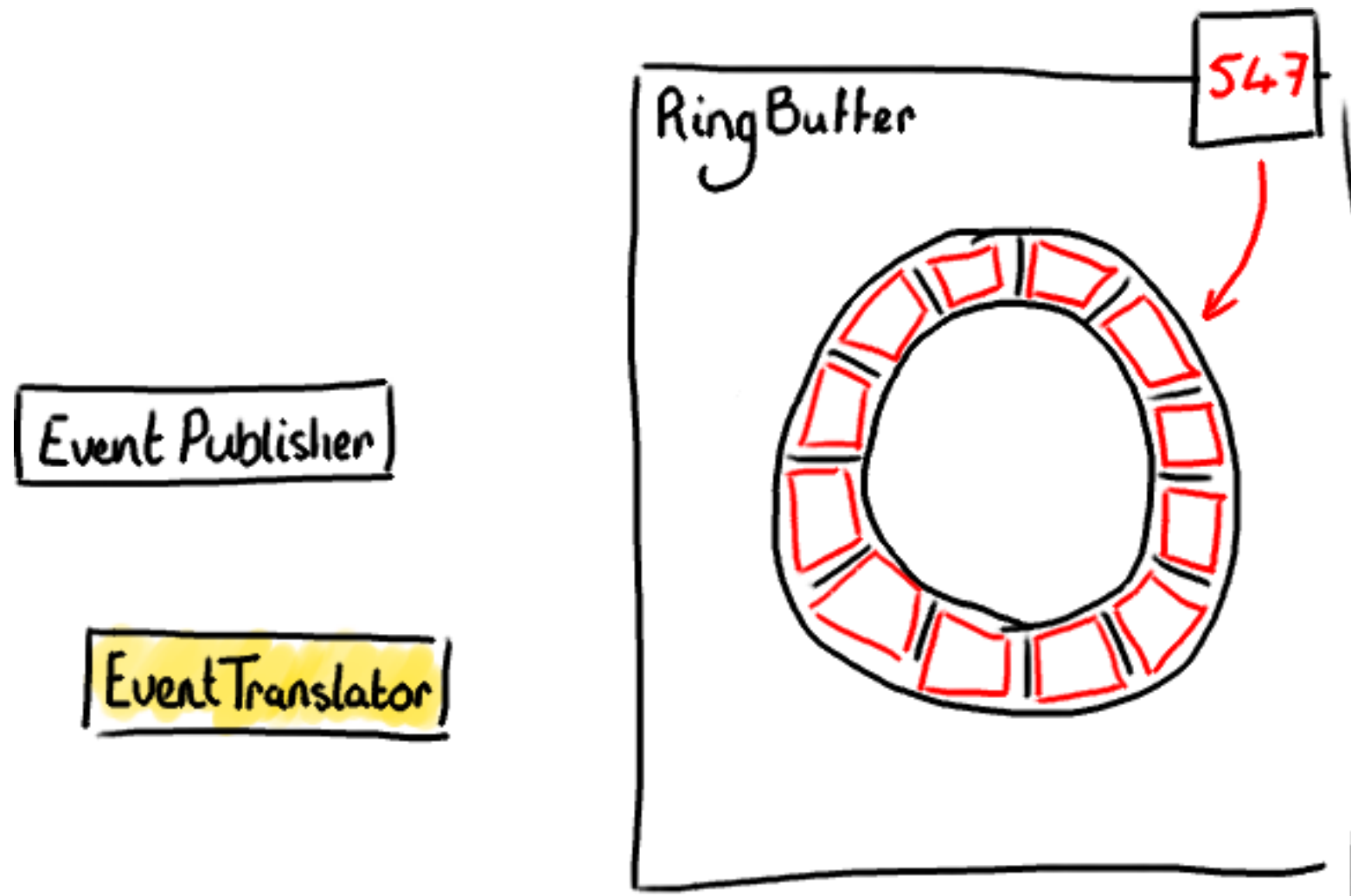
The Events are Buckets

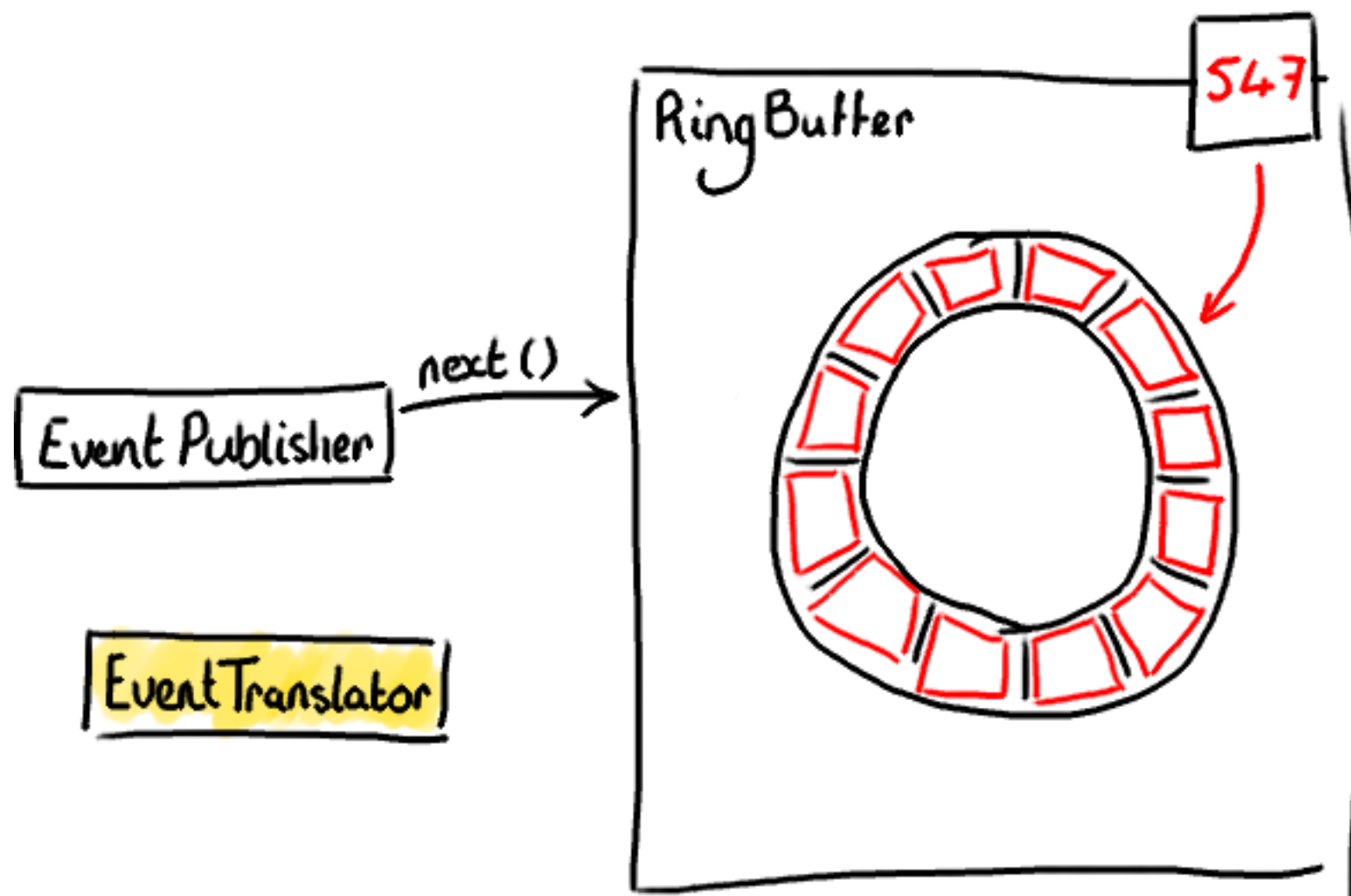


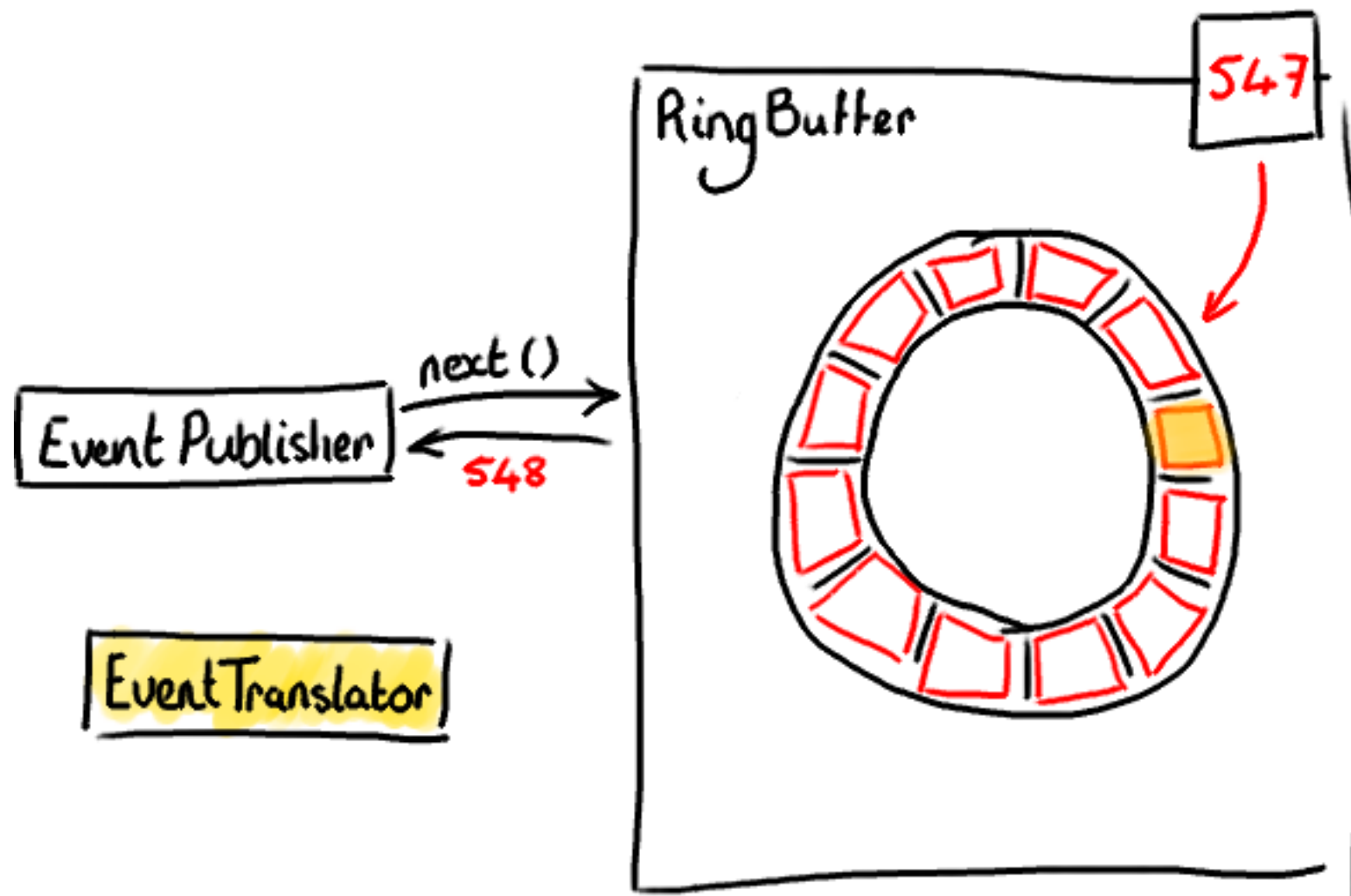
I've got a RingBuffer!

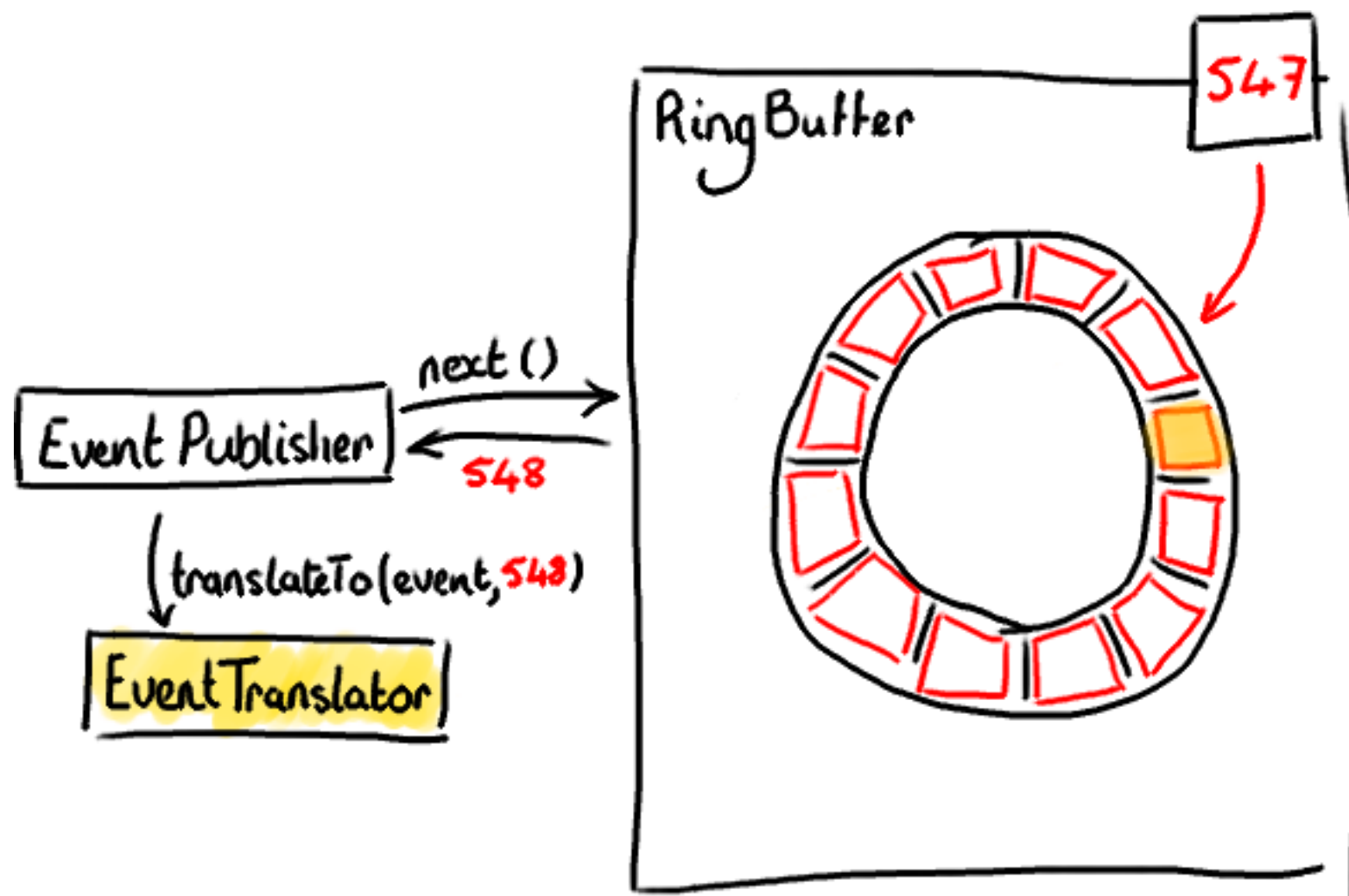
- Erm... how do I poke things into it?

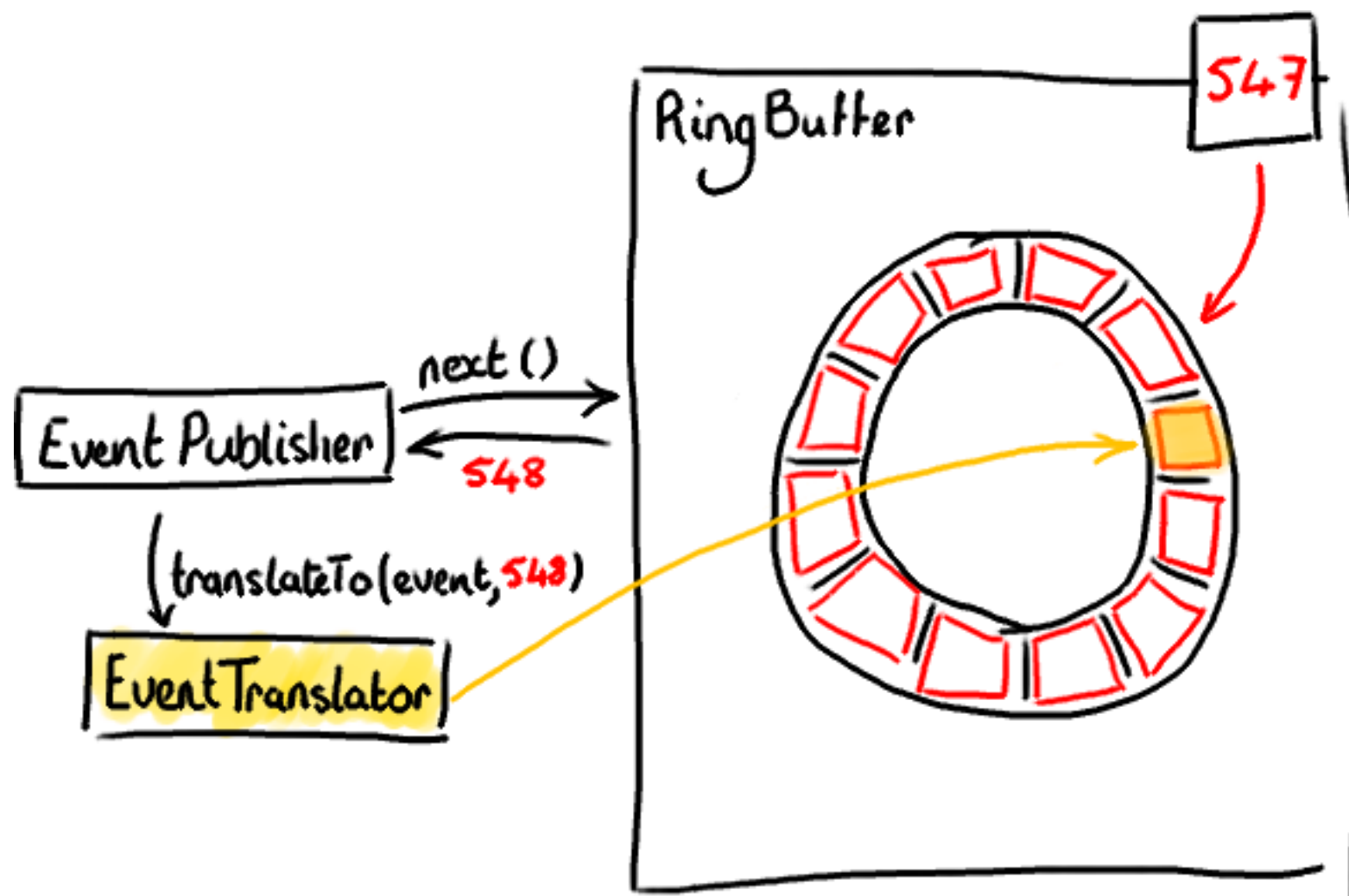
The Publisher

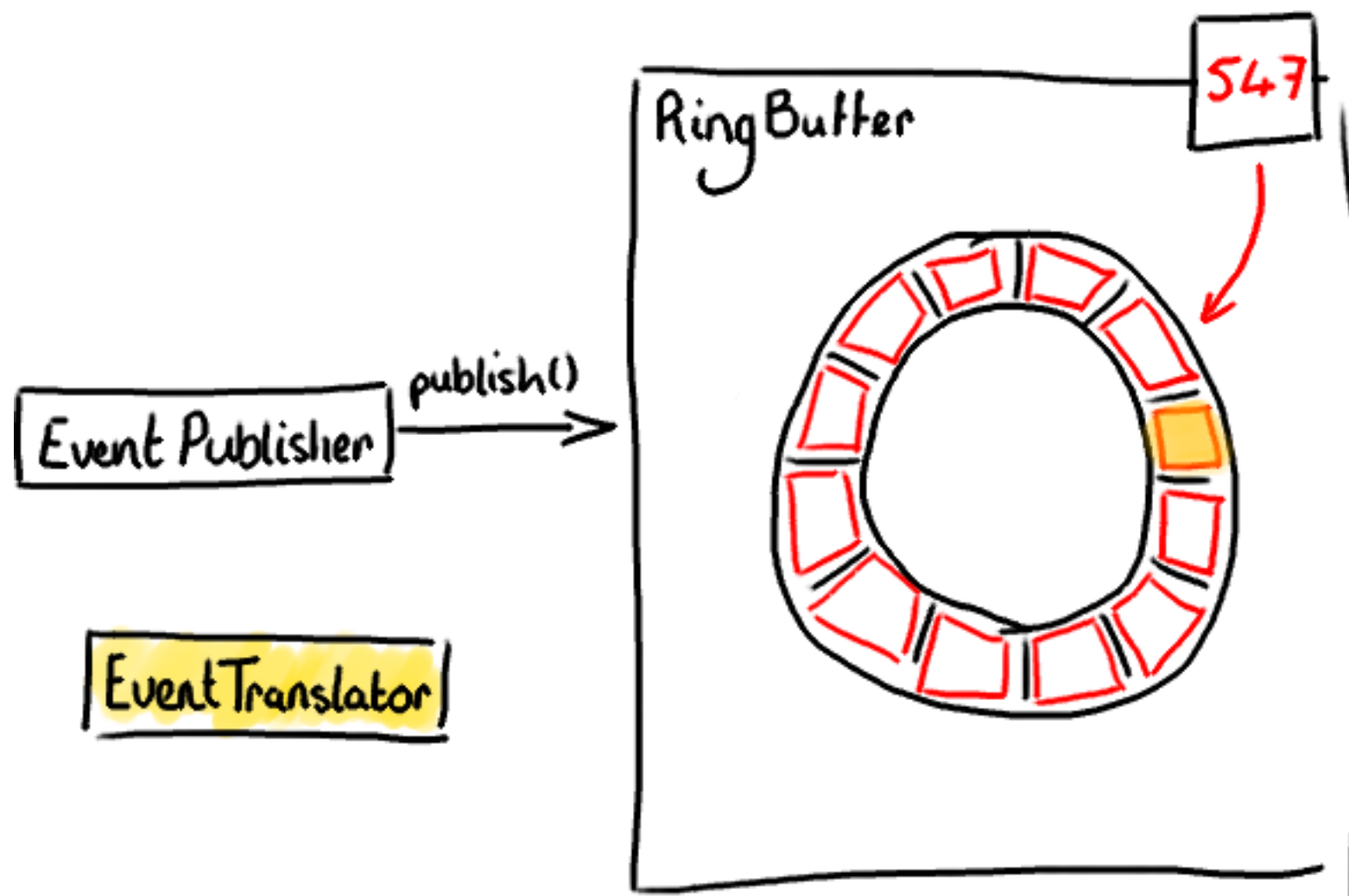


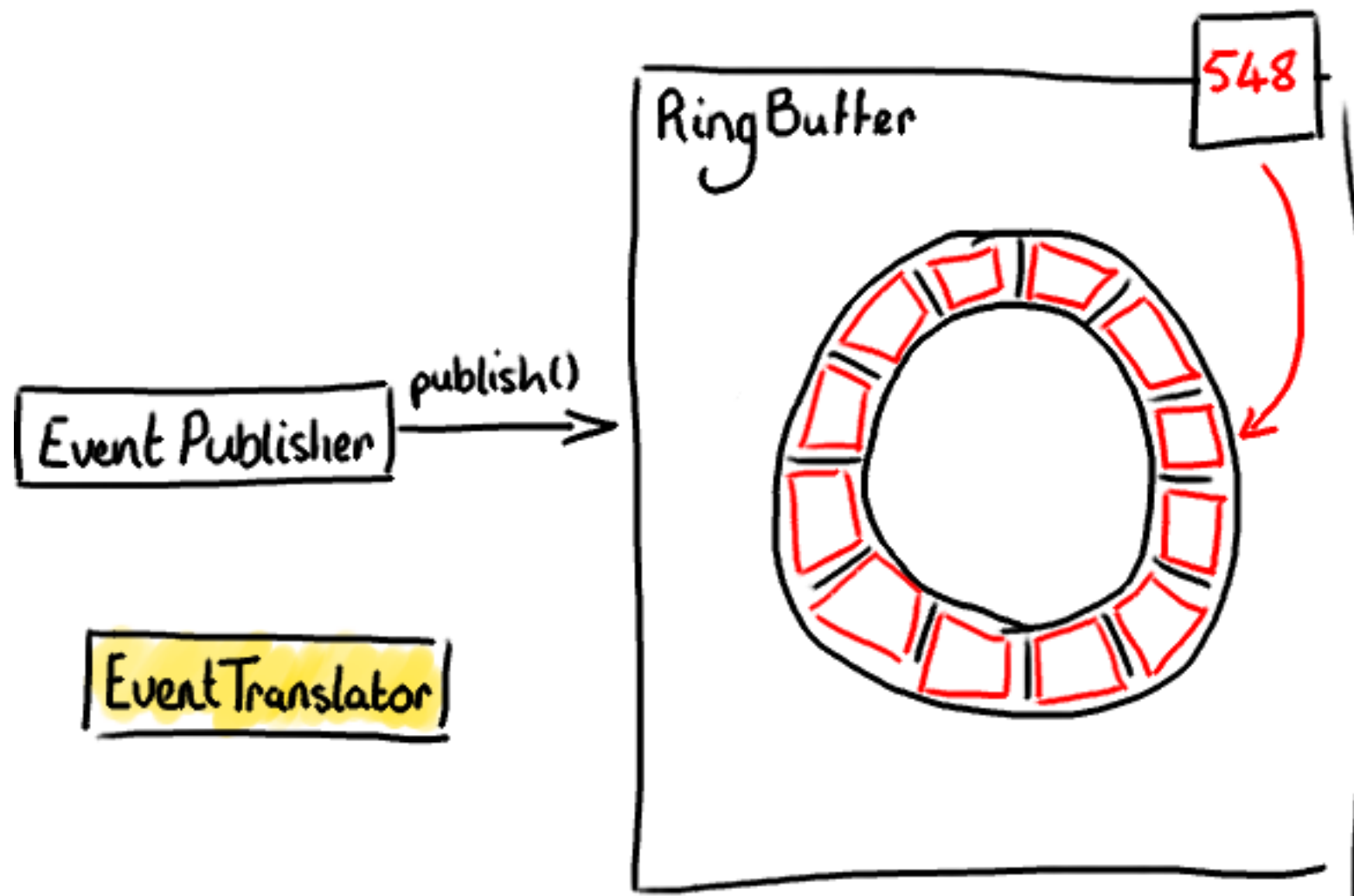








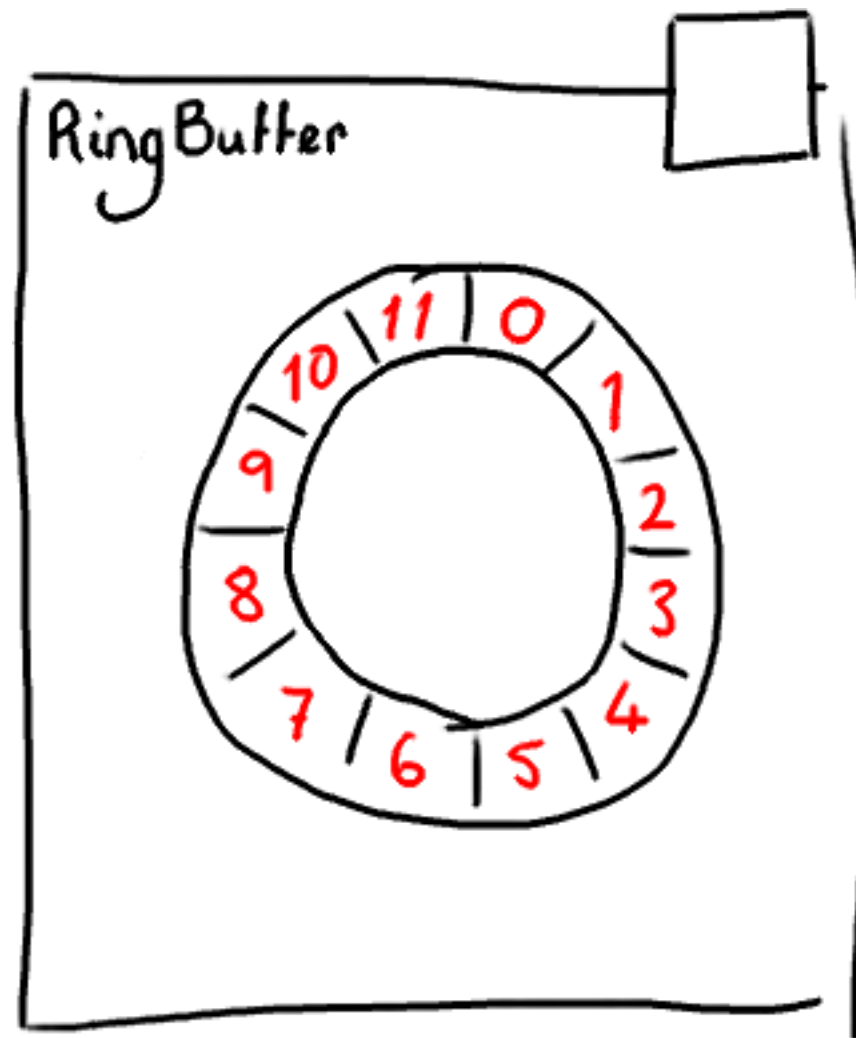


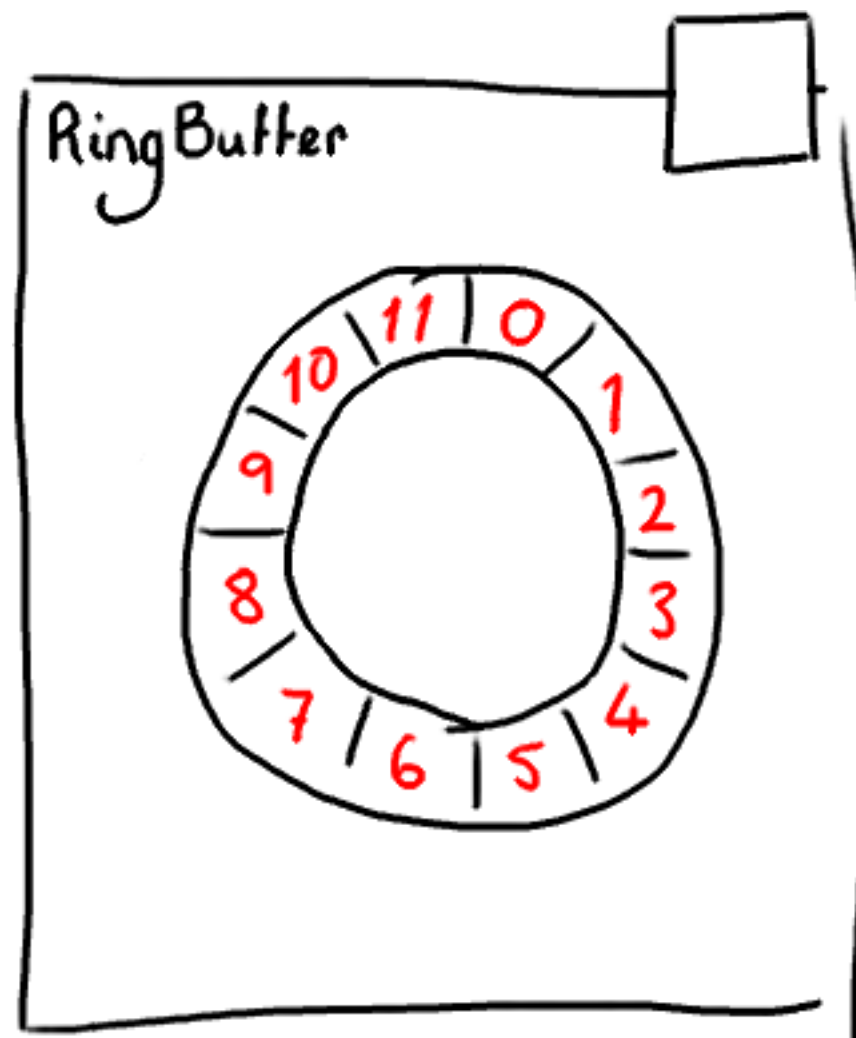


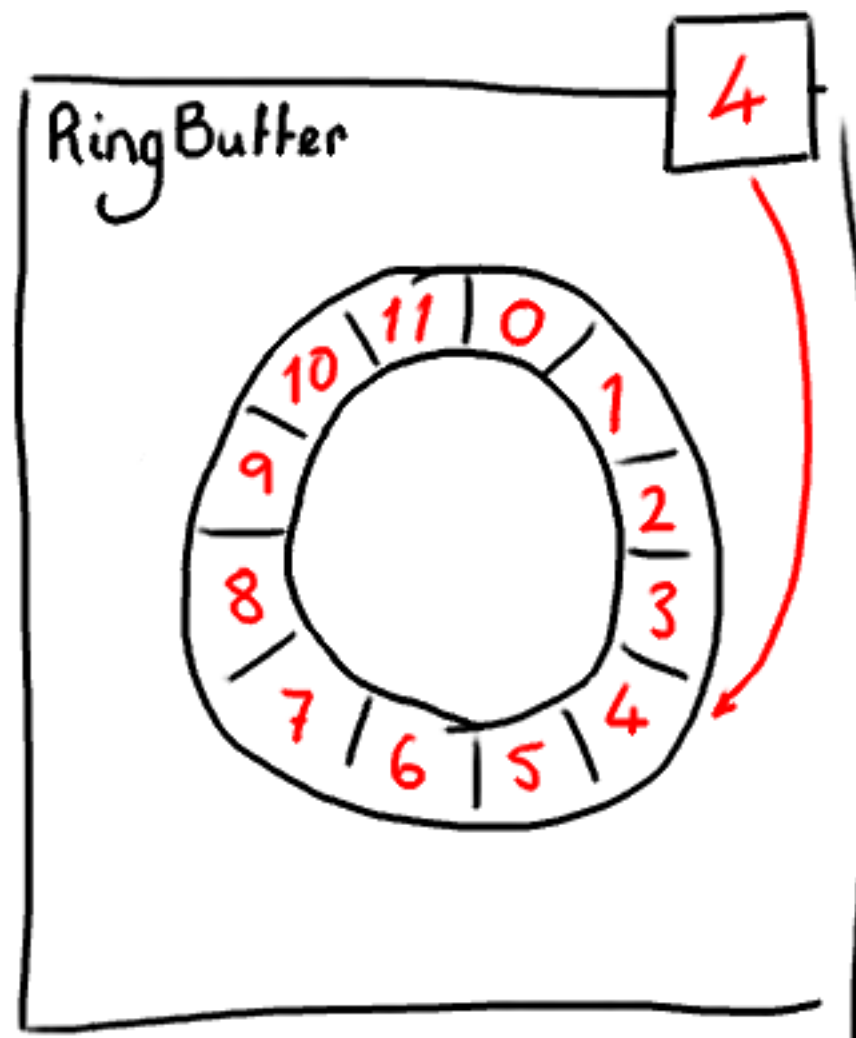
...so now I want to read

- The Disruptor provides nice batching behaviour for free

BatchEventProcessor

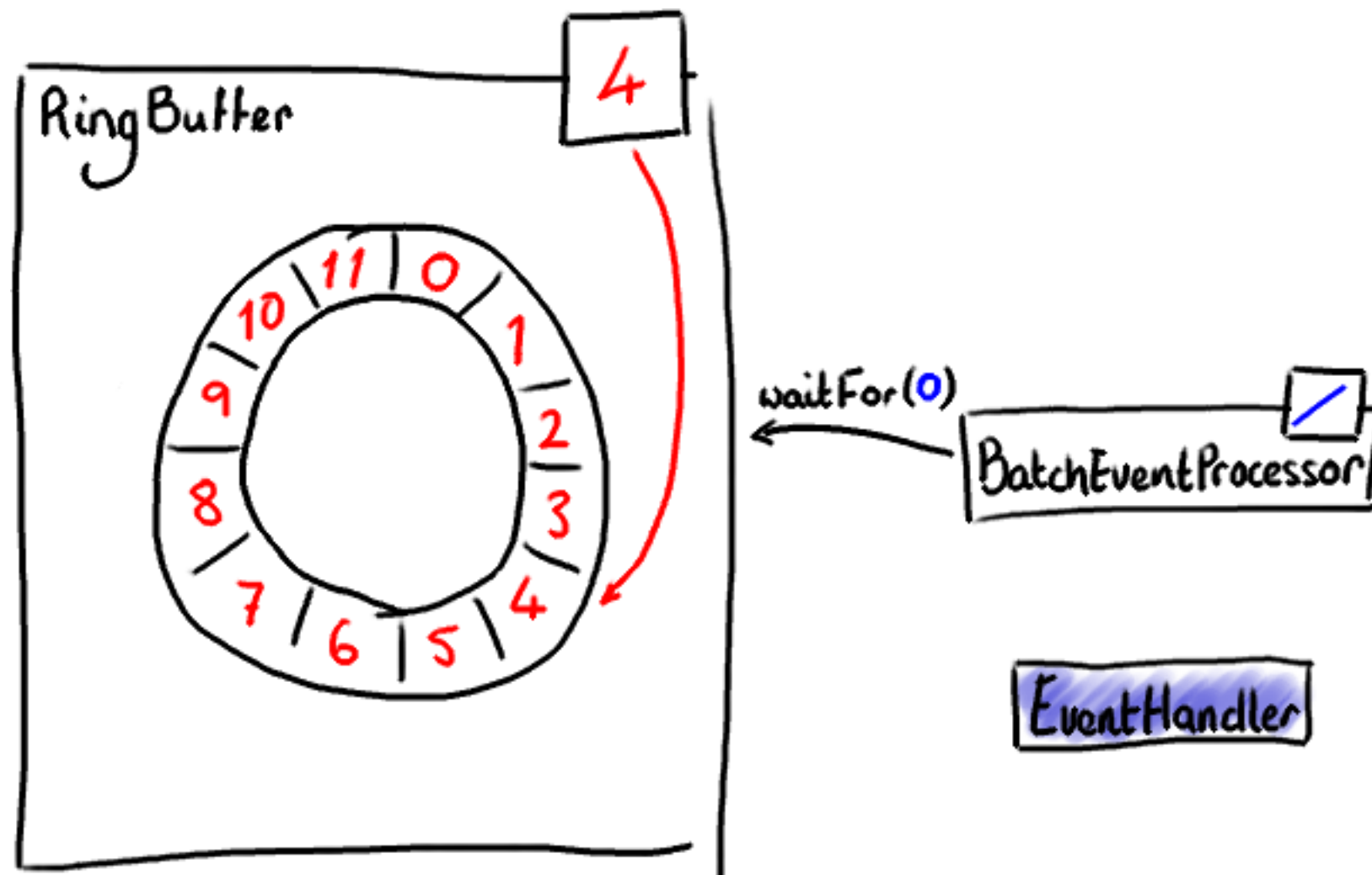


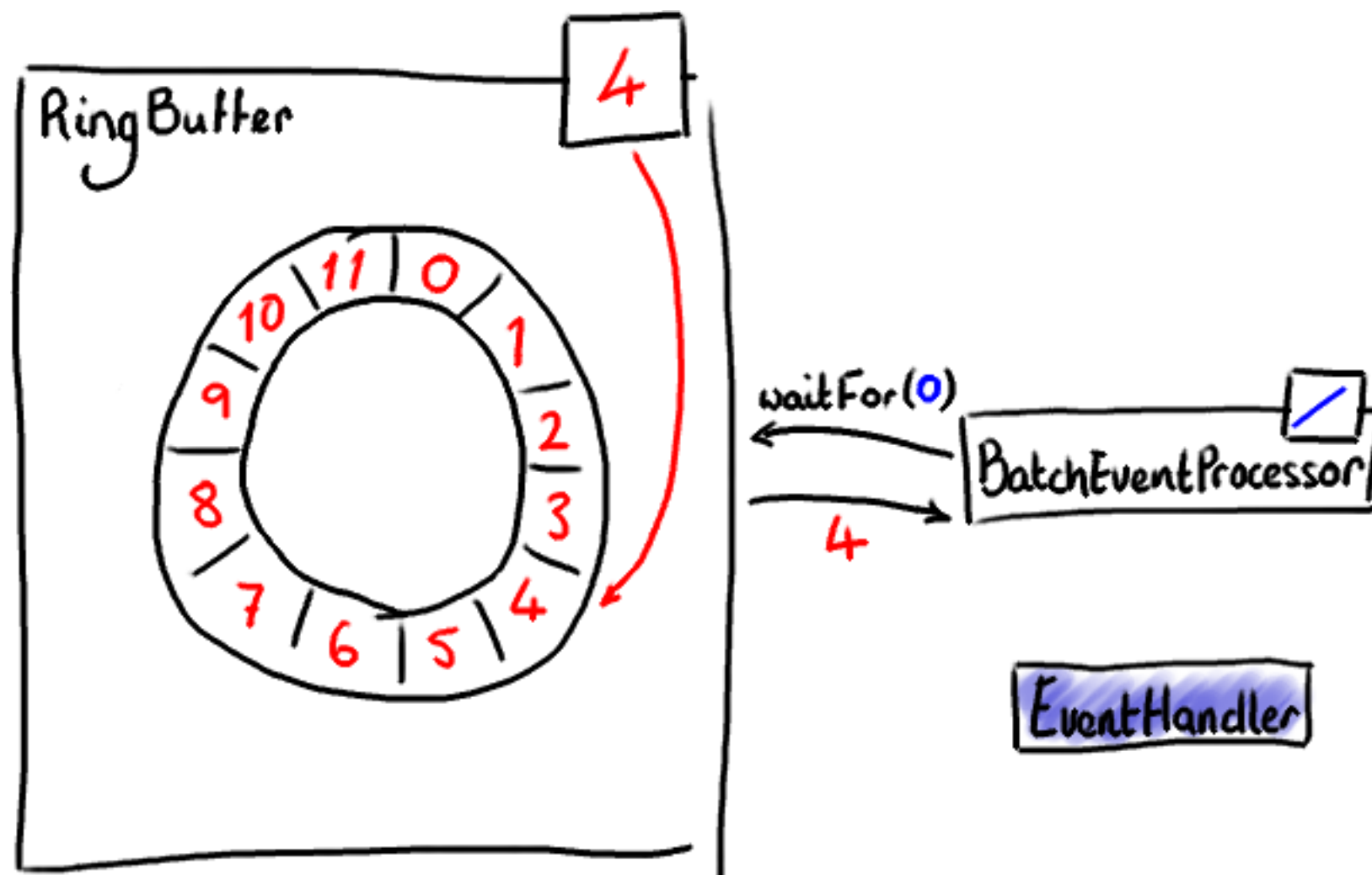


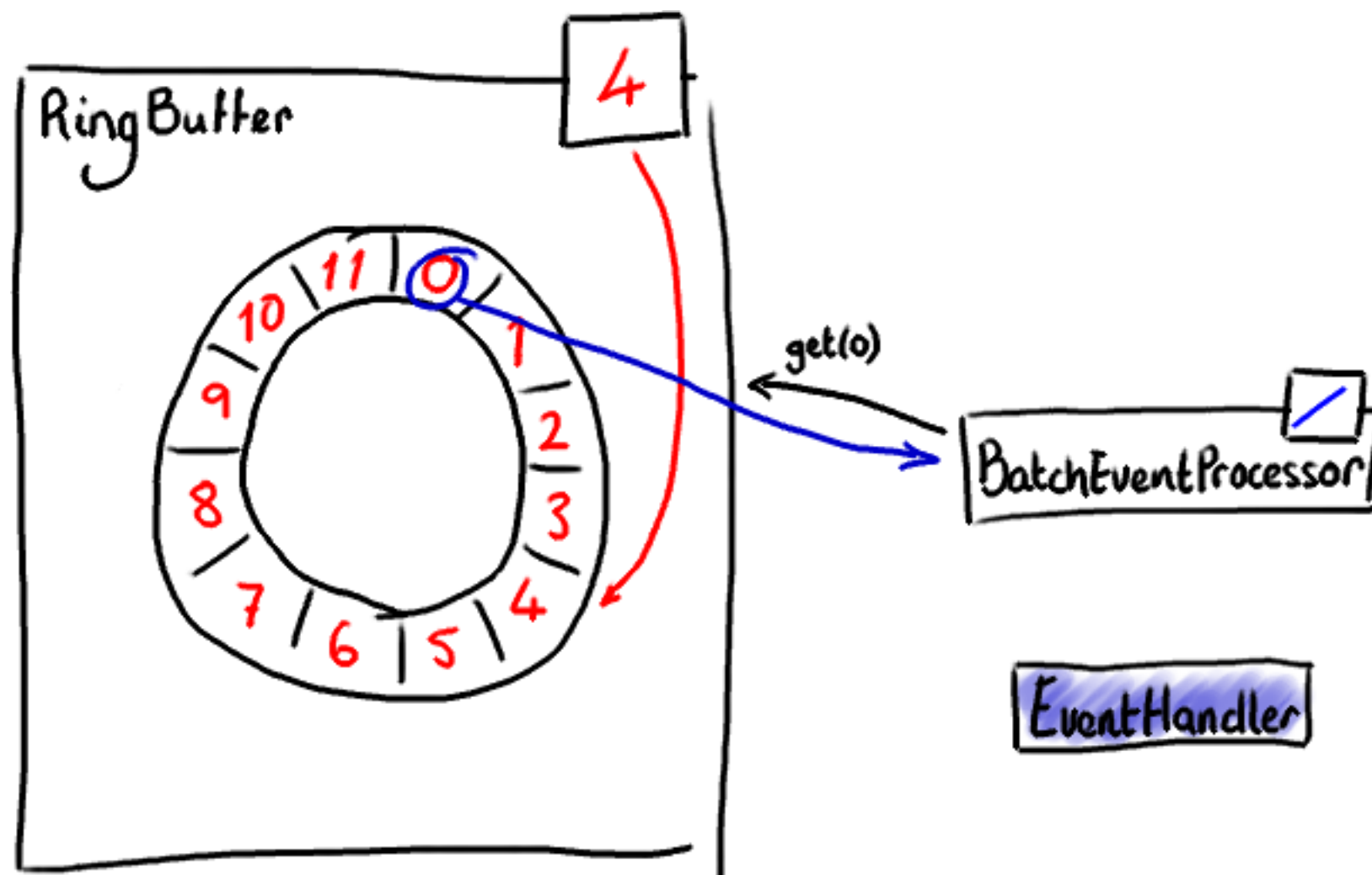


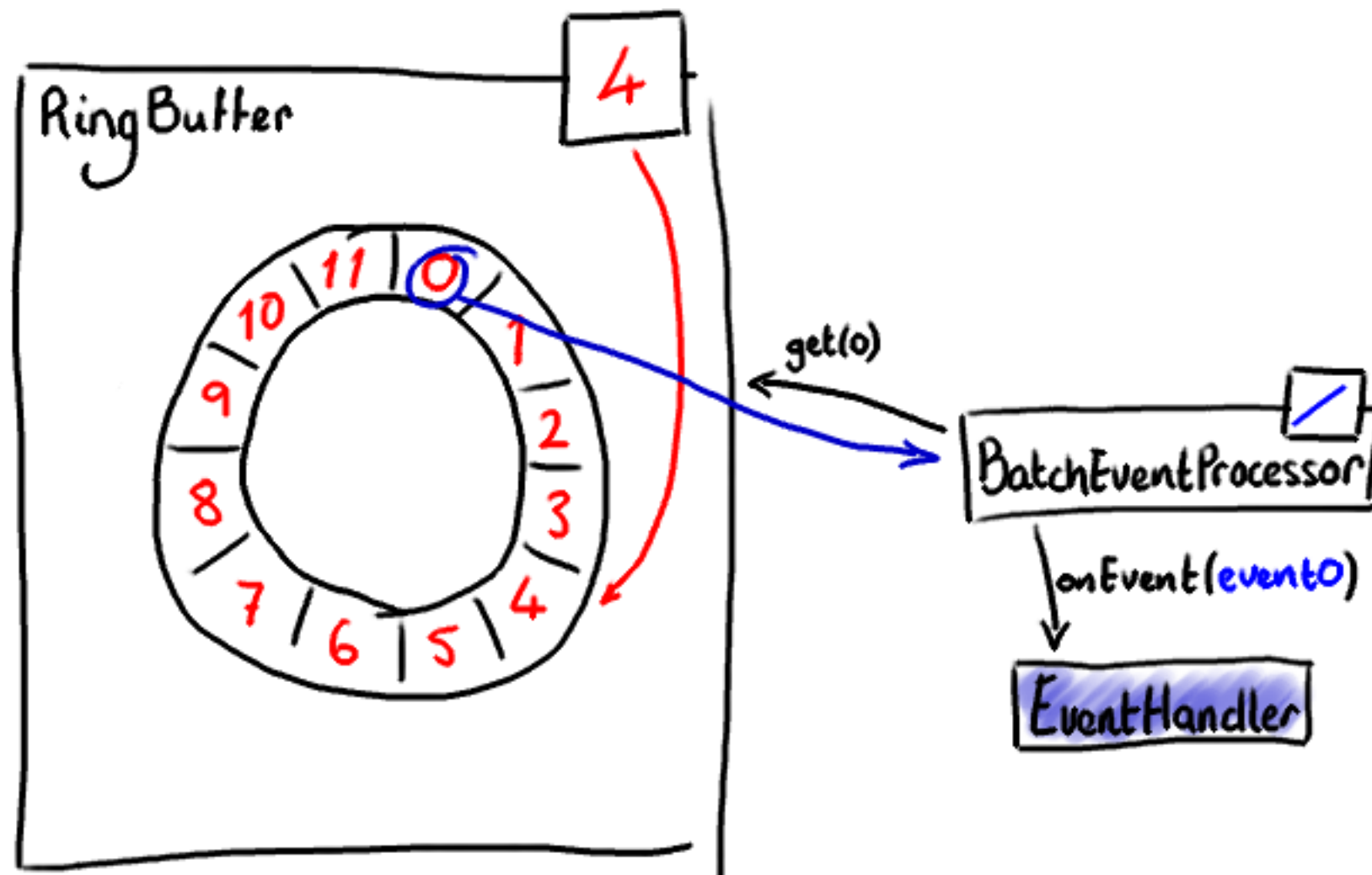
BatchEventProcessor

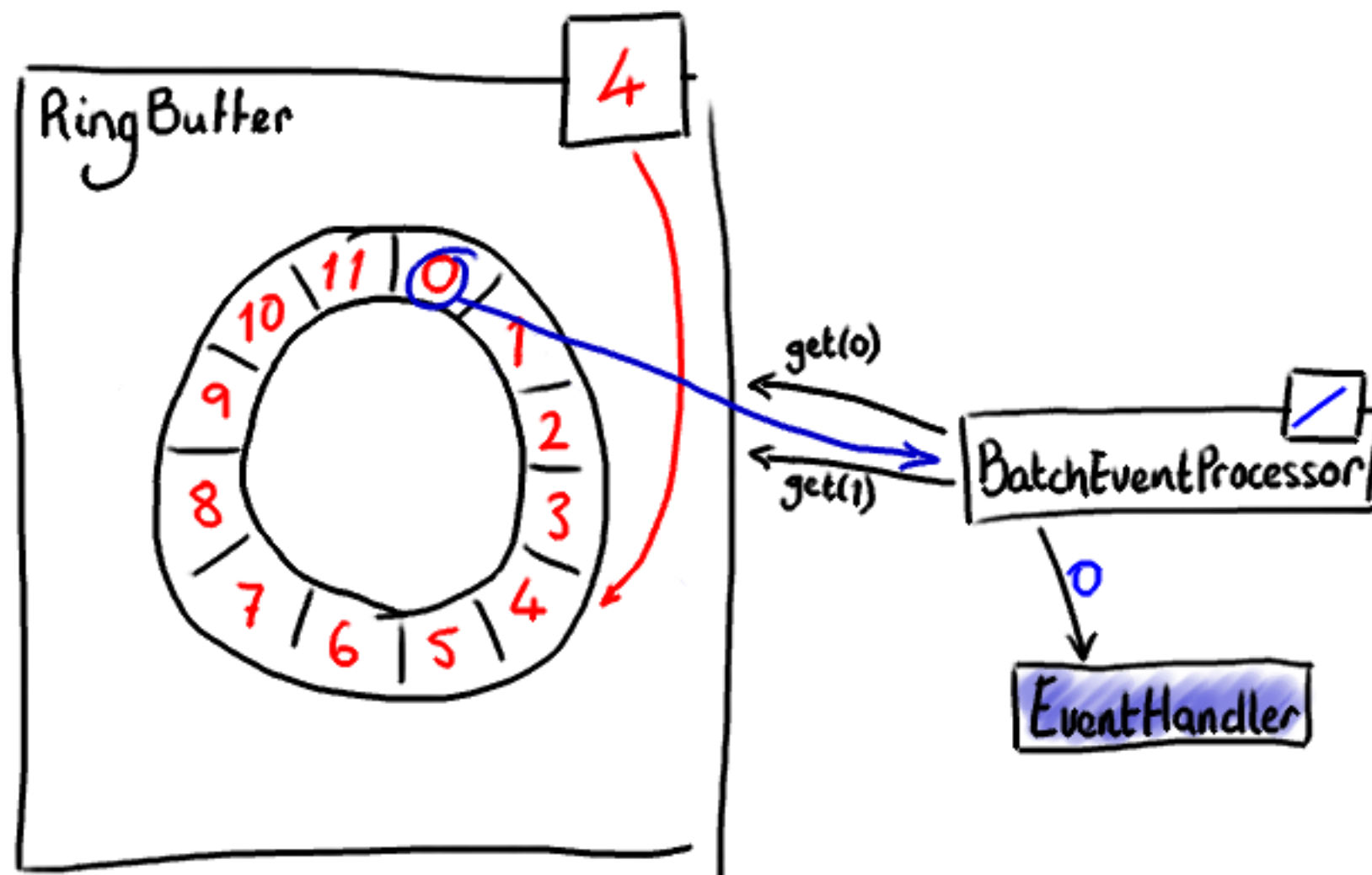
EventHandler

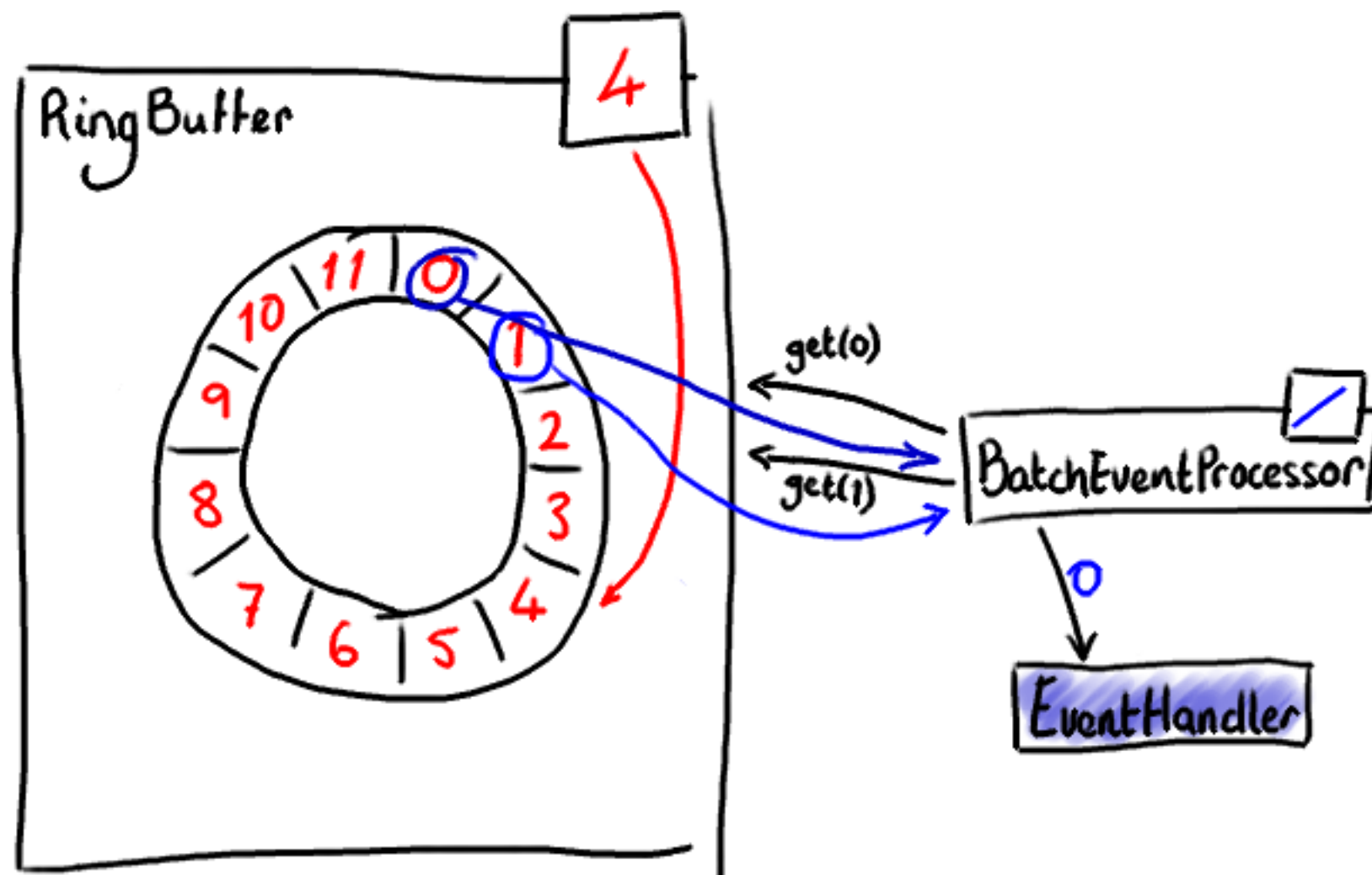


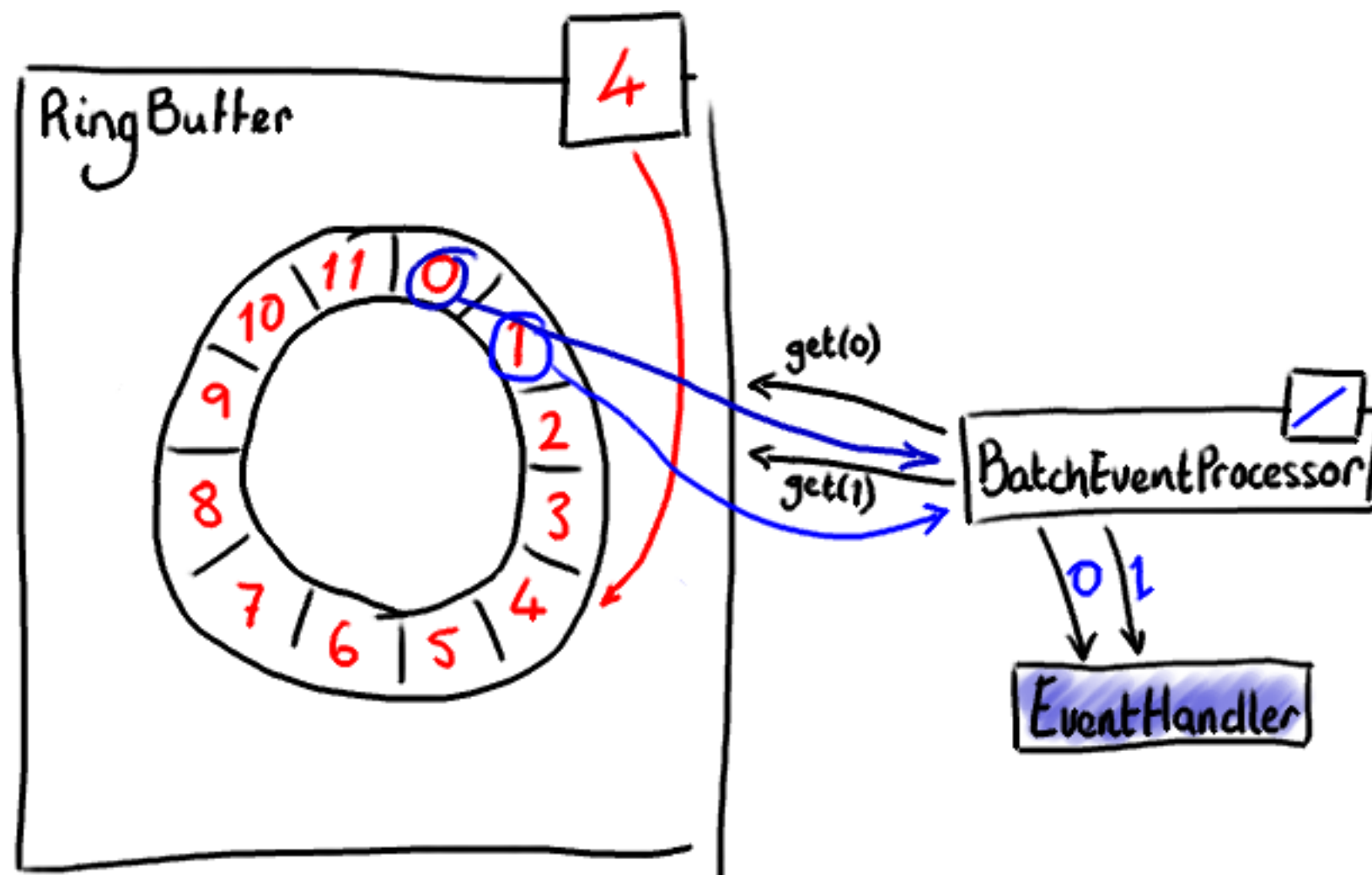


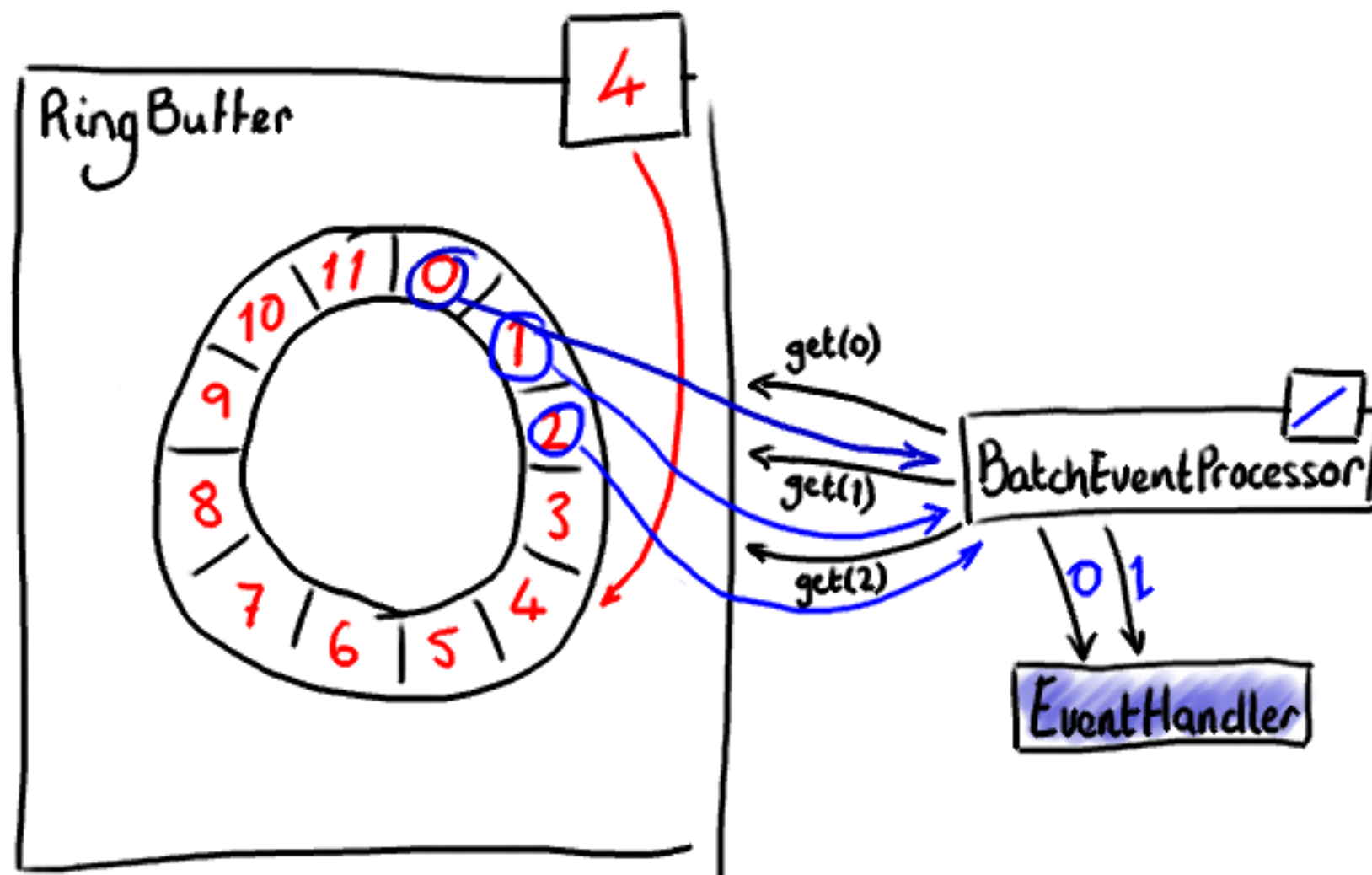


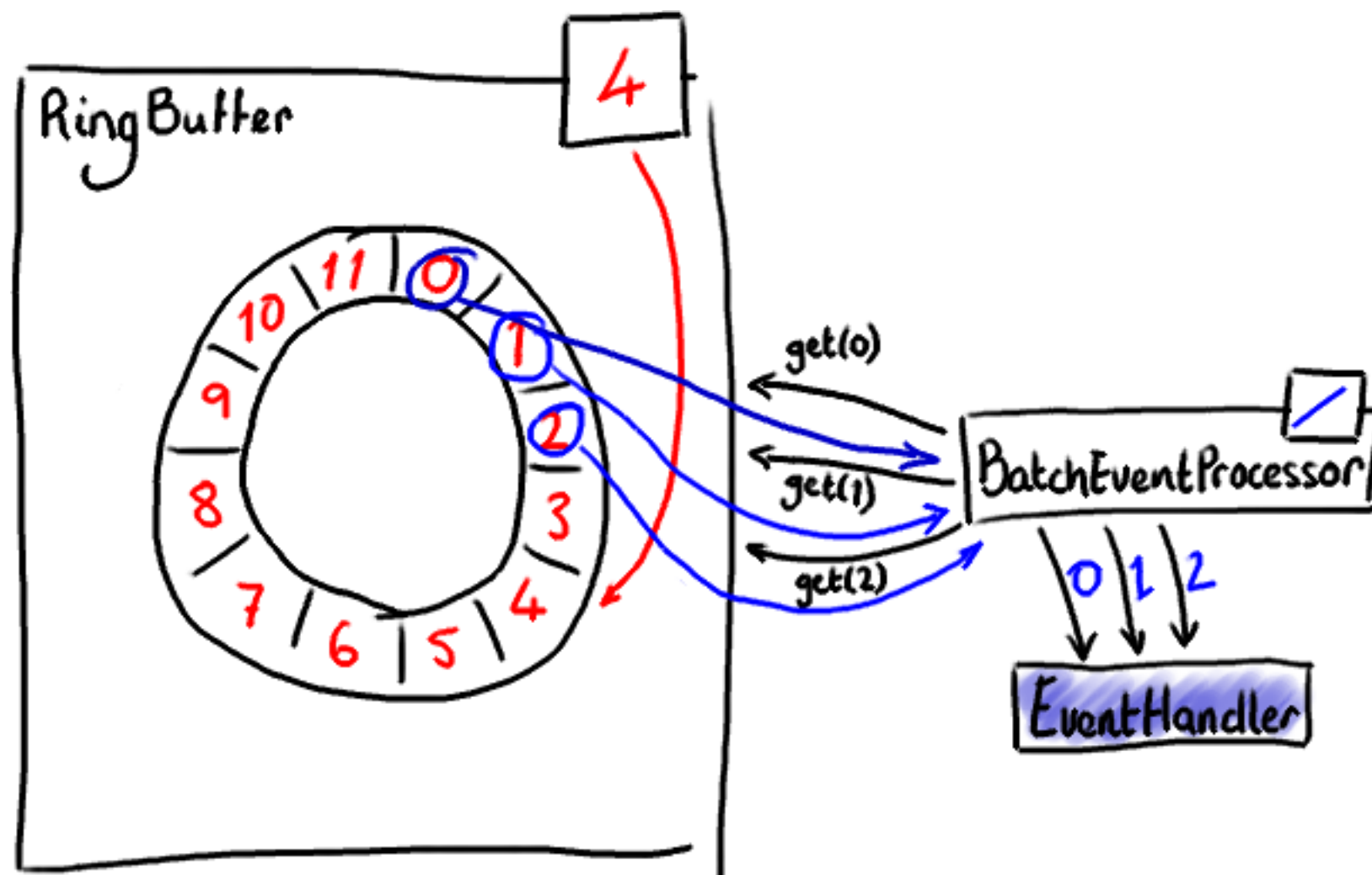


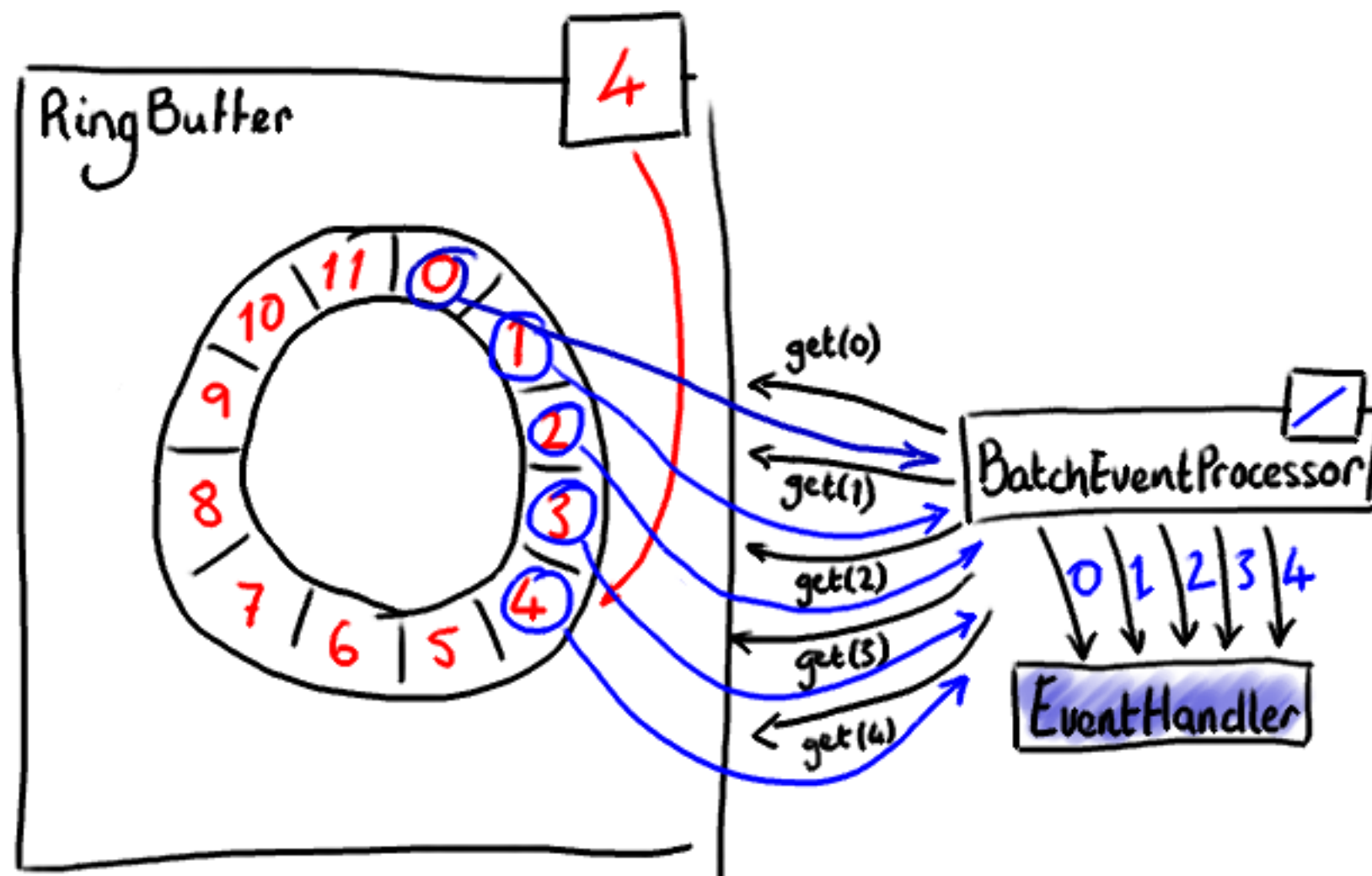


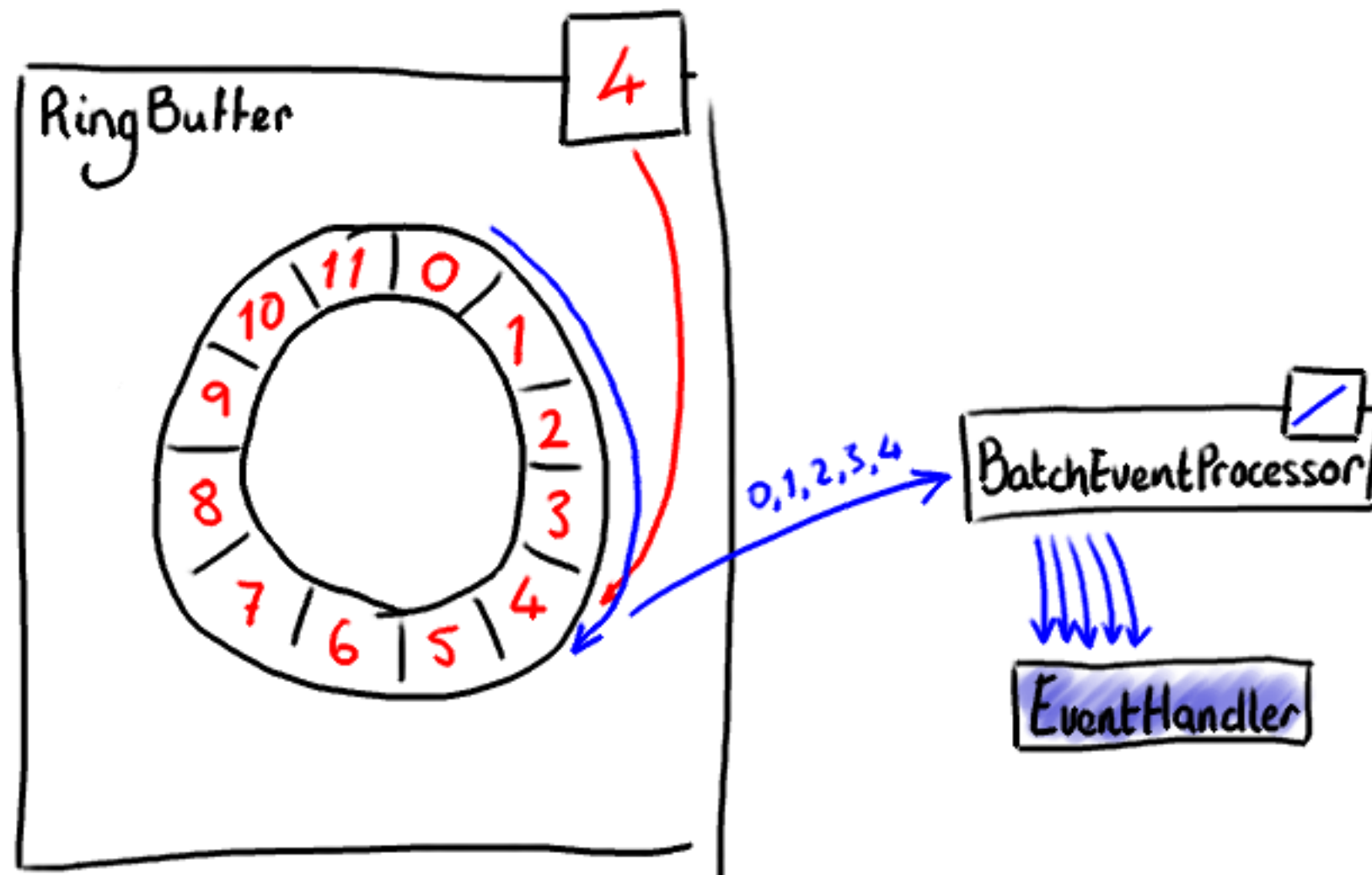


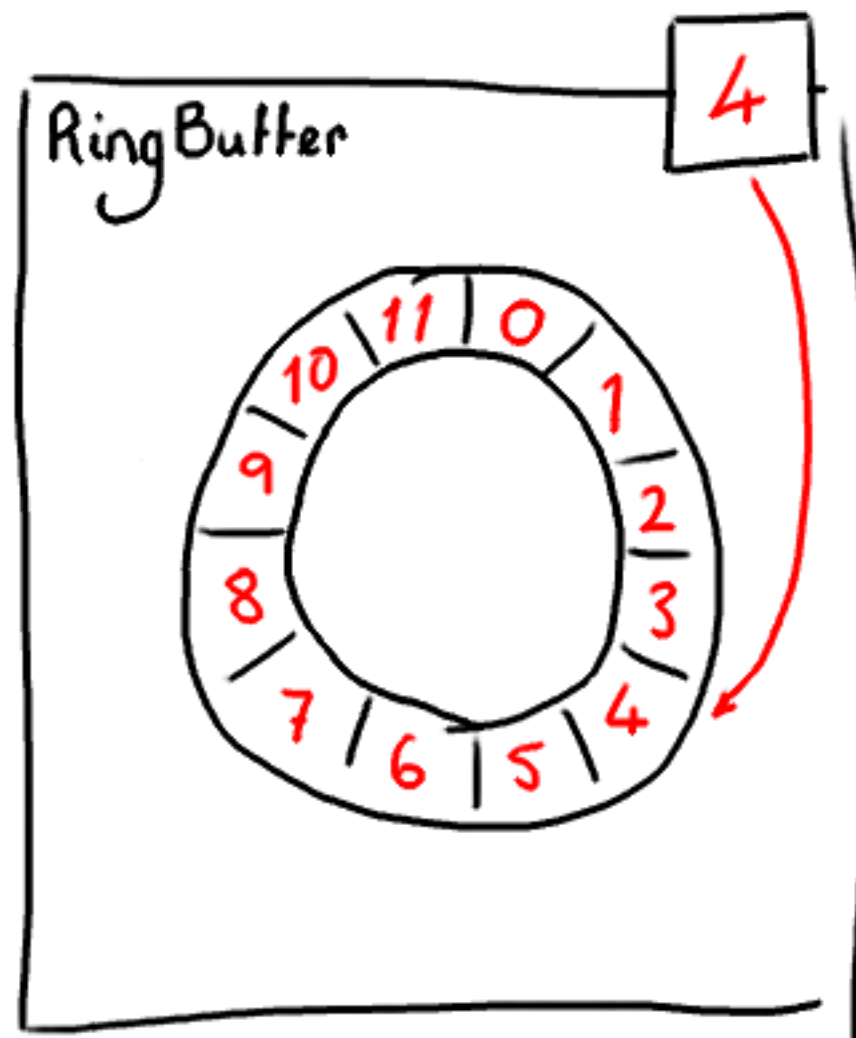






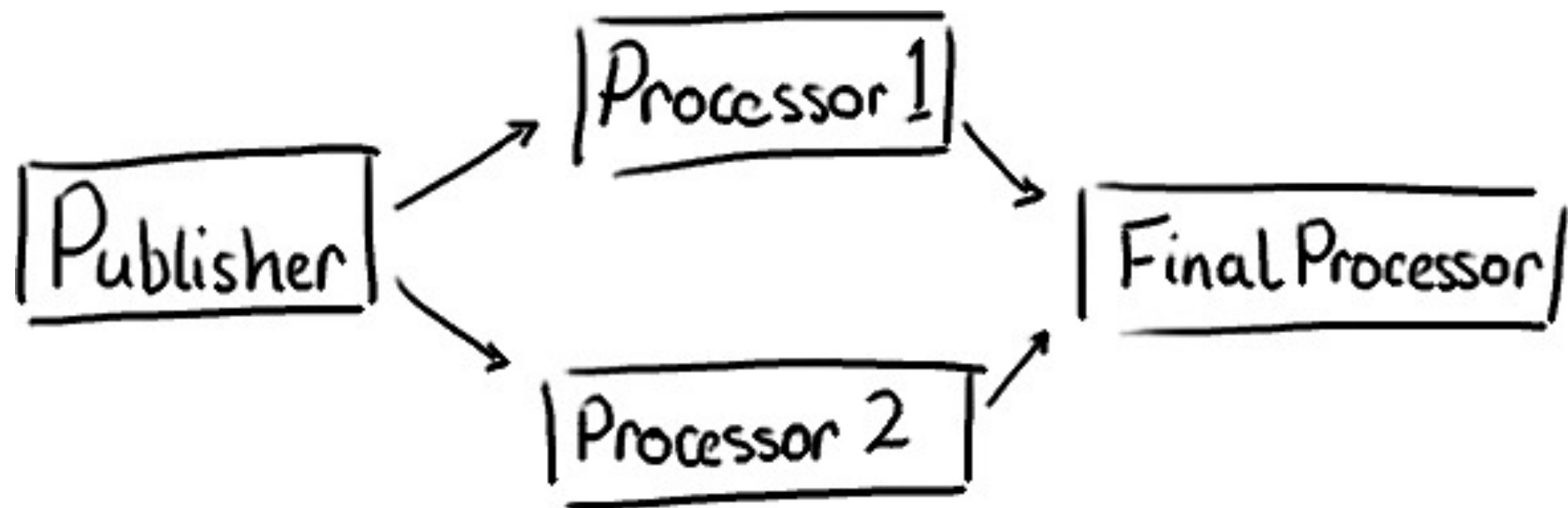




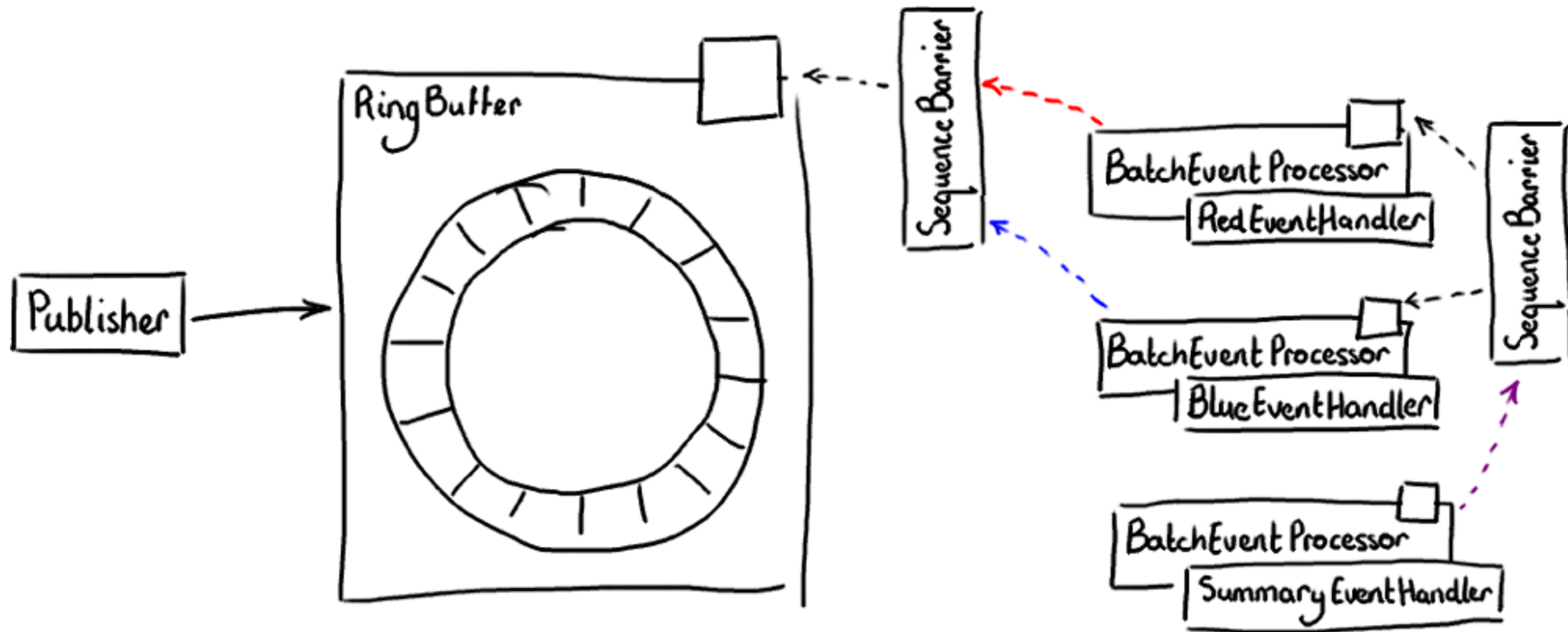


Shiny. So what?

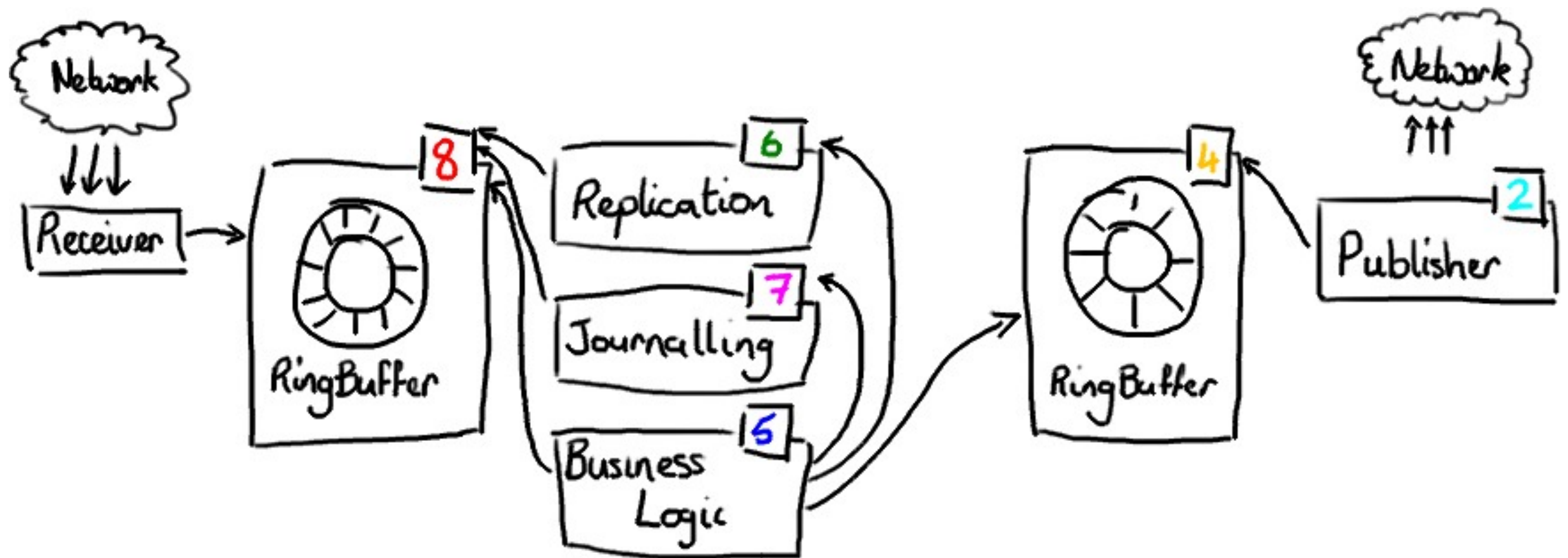
Complex Workflows



...and in the Disruptor?



Let's go parallel



Caveats

- Your ring buffer needs to be bigger than 12
- Event handlers are on separate threads
- Mileage May Vary - always performance test

You get...

- A framework that encourages you to model your domain
- The ability to run in parallel but single-threaded
- Nice, simple Java
- Reliable ordering
- ...and it can be very fast

Is that it?

- Wait strategies
- Batch publishing
- Multiple publishers
- The Wizard
- You don't even need a RingBuffer...

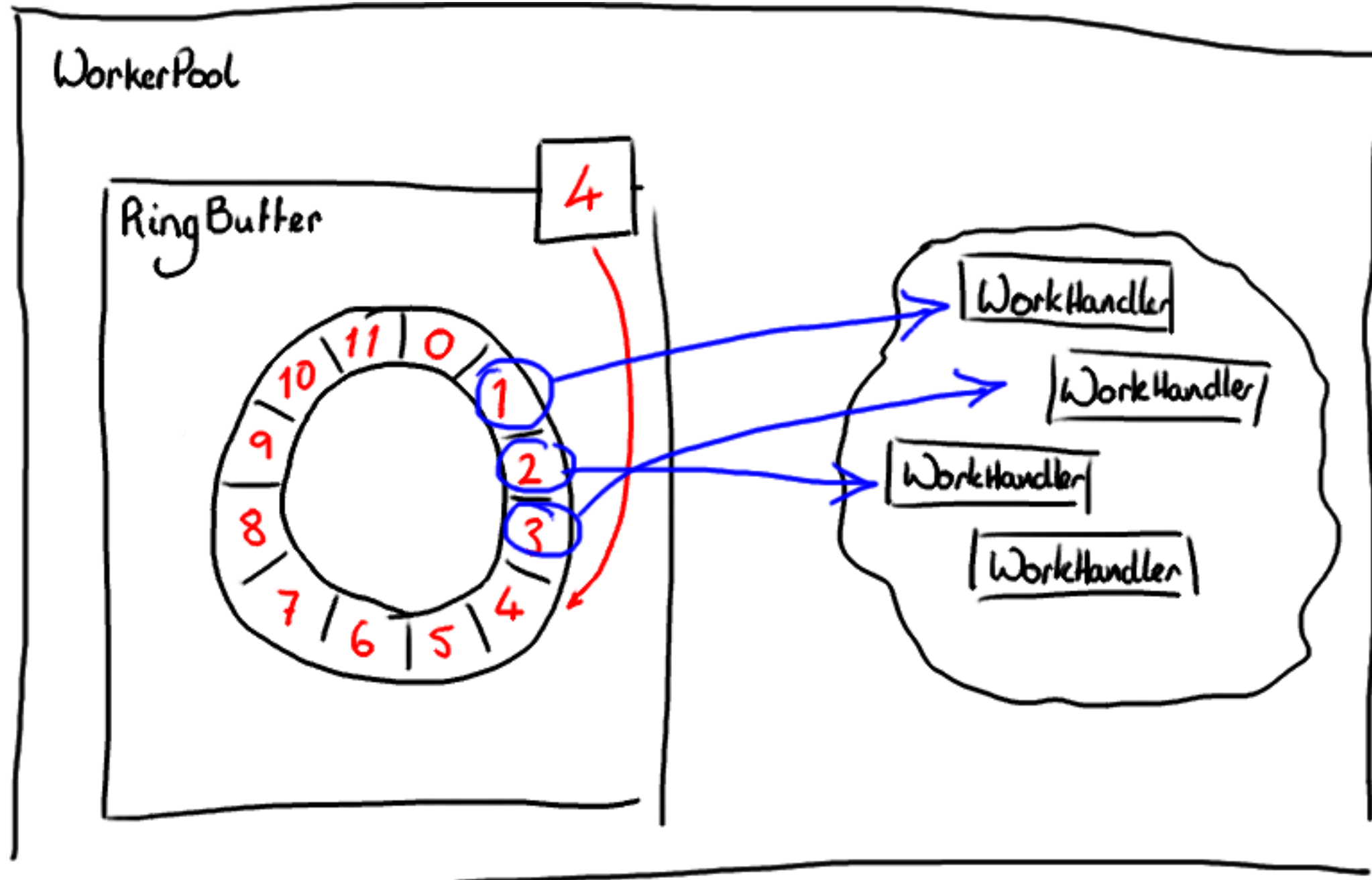
More Information

- Github:
github.com/LMAX-Exchange/disruptor
- Google Group
- Blogs

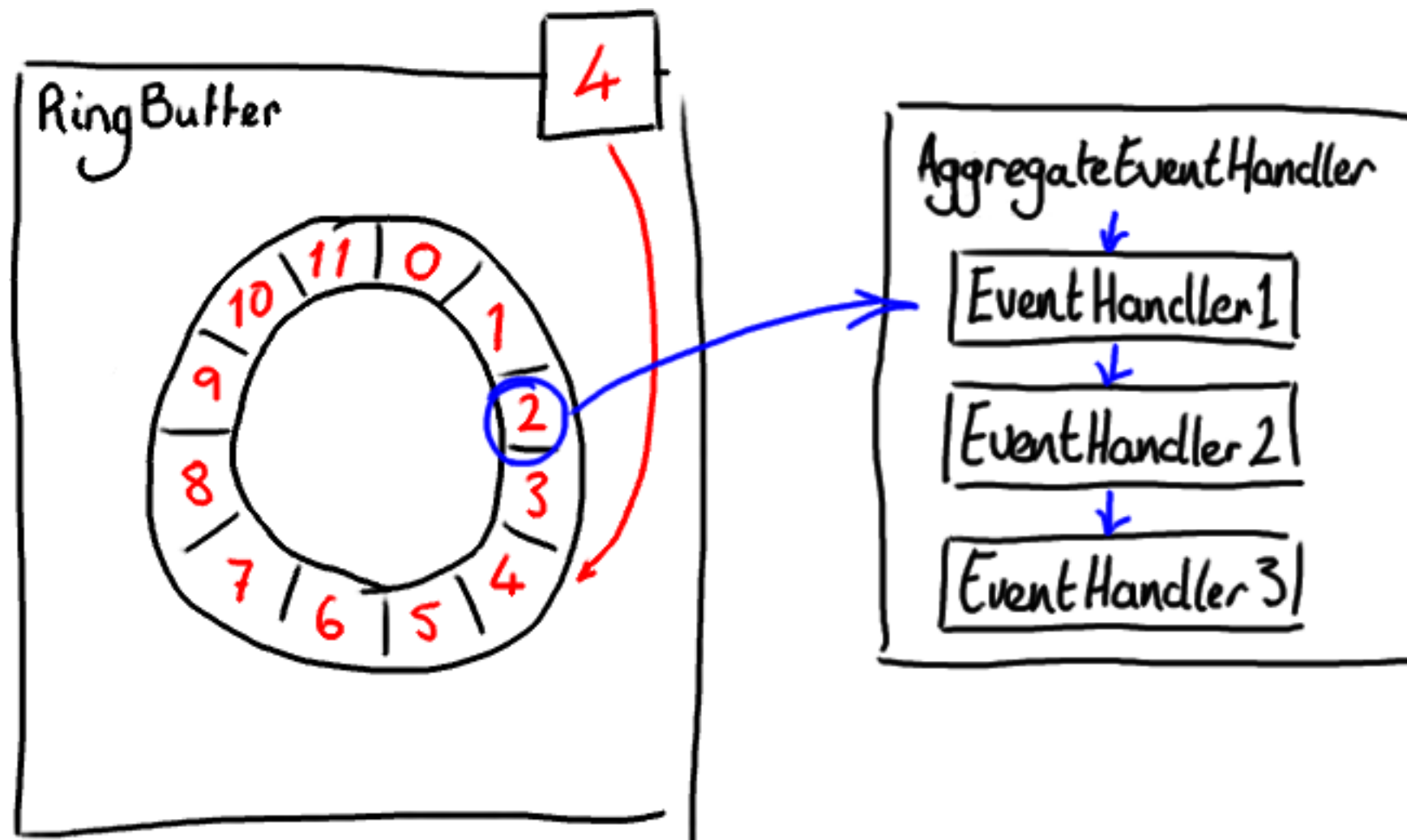
Q&A



WorkerPool



AggregateEventHandler



WaitStrategies

- BlockingWaitStrategy
- BusySpinWaitStrategy
- SleepingWaitStrategy
- YieldingWaitStrategy

ClaimStrategies

- SingleThreadedClaimStrategy
- MultiThreadedClaimStrategy
- MultiThreadedLowContentionClaimStrategy